MAGAZINE

BSD

FOR NOVICE AND ADVANCED USERS

Building a PCI Compliant Infrastructure on AWS

AWS Infrastructure Security

Deep Dive Into Access Control Management

UNIX BLOG PRESENTATION

Hubert Feyrer

Interview with Kalin Staykov

Elastix on Bhyve

Password Cracking in UNIX

Raspberry Pi 3

VOL. 11 NO. 03
ISSUE 03/2017 (91)
ISSN 1898-9144

FREENAS MINI
STORAGE APPLIANCE

IT *SAVES* YOUR LIFE.

HOW IMPORTANT IS YOUR DATA?

Years of family photos. Your entire music and movie collection. Office documents you've put hours of work into. Backups for every computer you own. We ask again, *how important is your data?*

NOW IMAGINE LOSING IT ALL

Losing one bit - that's all it takes. One single bit, and your file is gone.

The worst part? **You won't know until you absolutely need that file again.**

Example of one-bit corruption

THE SOLUTION

The FreeNAS Mini has emerged as the clear choice to save your digital life. **No other NAS in its class offers ECC (error correcting code) memory and ZFS bitrot protection to ensure data always reaches disk without corruption and *never degrades over time*.**

No other NAS combines the inherent data integrity and security of the ZFS filesystem with fast on-disk encryption. No other NAS provides comparable power and flexibility. The FreeNAS Mini is, hands-down, the best home and small office storage appliance you can buy on the market. **When it comes to saving your important data, there simply is no other solution.**

The Mini boasts these state-of-the-art features:

- 8-core 2.4GHz Intel® Atom™ processor
- Up to 16TB of storage capacity
- 16GB of ECC memory (with the option to upgrade to 32GB)
- 2 x 1 Gigabit network controllers
- Remote management port (IPMI)
- Tool-less design; hot swappable drive trays
- FreeNAS installed and configured

iXsystems™

http://www.iXsystems.com/mini

intel® inside™ ATOM™

FREENAS CERTIFIED STORAGE

With over six million downloads, FreeNAS is undisputedly *the* most popular storage operating system in the world.

Sure, you could build your own FreeNAS system: research every hardware option, order all the parts, wait for everything to ship and arrive, vent at customer service because it *hasn't*, and finally build it yourself while hoping everything fits - only to install the software and discover that the system you spent *days* agonizing over **isn't even compatible**. Or...

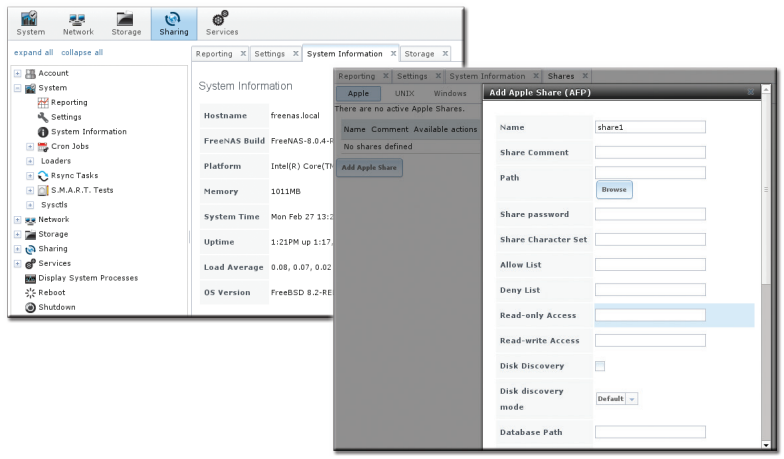
MAKE IT EASY ON YOURSELF

As the sponsors and lead developers of the FreeNAS project, iXsystems has combined over 20 years of hardware experience with our FreeNAS expertise to bring you FreeNAS Certified Storage. **We make it easy to enjoy all the benefits of FreeNAS without the headache of building, setting up, configuring, and supporting it yourself.** As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS.

Every FreeNAS server we ship is...

» Custom built and optimized for your use case
» Installed, configured, tested, and guaranteed to work out of the box
» Supported by the Silicon Valley team that designed and built it
» Backed by a 3 years parts and labor limited warranty

As one of the leaders in the storage industry, you know that you're getting the best combination of hardware designed for optimal performance with FreeNAS. **Contact us today for a FREE Risk Elimination Consultation with one of our FreeNAS experts.** Remember, every purchase directly supports the FreeNAS project so we can continue adding features and improvements to the software for years to come. **And really - why would you buy a FreeNAS server from *anyone* else?**



FreeNAS 1U

- Intel® Xeon® Processor E3-1200v2 Family
- Up to 16TB of storage capacity
- 16GB ECC memory (upgradable to 32GB)
- 2 x 10/100/1000 Gigabit Ethernet controllers
- Redundant power supply

FreeNAS 2U

- 2x Intel® Xeon® Processors E5-2600v2 Family
- Up to 48TB of storage capacity
- 32GB ECC memory (upgradable to 128GB)
- 4 x 1GbE Network interface (Onboard) - (Upgradable to 2 x 10 Gigabit Interface)
- Redundant Power Supply

http://www.iXsystems.com/storage/freenas-certified-storage/

Editor's Words

Dear Readers,

We selected security as the main topic for this BSD issue because it is one of the most often repeated needs nowadays. We all want security, not only in our virtual world, but also in our real-life situations. The meaning of security is different for everyone. This can be related to different needs, rules and ideas. If we talk about real life, security is important no matter what it really means and what kinds of conditions need to be fulfilled. This kind of security depends on real needs, and it can be described by many different rules depending on country, nationality, law, or even mood. When I think about IT security, which of course is one of the most important factors in making us feel secure in real life dimension, I think about standards, procedures and technology. And this BSD issue will present you with more information about them.

The article I will recommend is titled B*uilding a PCI Compliant Infrastructure on AWS* by Renan Dias. He will be building a quite complex AWS infrastructure and he will be deploying an application called Guestbook. He claims that everything you will learn can be applied to any infrastructure running any application. The article is long, but I decided not to share it into two issues to make your life easier and give you the full guidelines on how to secure yourself. I hope you enjoy reading it and start building your own infrastructure on AWS.

The next article will help you secure your IT world. It is *AWS Infrastructure Security: Deep Dive into Access Control Management* by Mohamed Farag. In his article, Mohamed introduces you to various considerations with access control management in AWS infrastructure. You will learn different tips and tricks to think through the security of your access control management and exploit a wide-variety of tools to improve your organization's security infrastructure. And do not omit the *Password Cracking in UNIX* article by Amit Chugh. I hope you will find some more support for your individual projects.

If you want to start with Raspberry Pi and don't know how, just go to the next pages to read the article titled *Ready to Land on IoT World with the Raspberry Pi 3* by Manuel Daza.

The next article worth reading is *Elastix On Bhyv*e by Abdorrahman Homaei. I would like to add that Abdorrahman Homaei's articles will be published by the BSD magazine more regularly. Therefore, if you like his writing, please contact him to give some feedback for future articles.

Since I am limited by words for this editorial, please see the Table of Contents for a more thorough description on the articles.

We also have a new blog presentation. I think that you know him very well. This month you can meet Hubert Feyrer.

Of course, please do not forget to see our interview for this issue. This time Kalin Staykov responds to our questions. Check out what he thinks about DevOps.

Also, I would like to invite you to read the column by Rob Somerville. It is always a good read.

As long as we have our dear readers, we have a purpose. We owe you a huge THANK YOU, Additionally, we are grateful for every comment and opinion, whether positive or negative. Every word from you compels us to improve the BSD magazine and brings us closer to the ideal shape of our publication.

Best regards,
Ewa & The BSD Team

PS. If you want to start a real life open-source journey with our rich-content publications, or if you want to get in contact with our team, feel free to write to us.

Table of Contents

NetBSD 7.1 Released

The NetBSD Project announced that NetBSD 7.1, the first feature update of the NetBSD 7 release branch, is available to download. It represents a selected subset of fixes deemed important for security or stability reasons, as well as new features and enhancements. Some highlights of the 7.1 release are:

- Support for Raspberry Pi Zero.
- Initial DRM/KMS support for NVIDIA graphics cards via nouveau (Disabled by default. Uncomment nouveau and nouveaufb in your kernel config to test).
- The addition of vioscsi, a driver for the Google Compute Engine disk.
- Linux compatibility improvements, allowing, e.g., the use of Adobe Flash Player 24.
- wm(4)
- ODROID-C1 Ethernet now works.
- Numerous bug fixes and stability improvements.

Complete source and binaries for NetBSD 7.1 are available for download at many sites around the world. A list of download sites providing FTP, AnonCVS, SUP and other services may be found at http://www.NetBSD.org/mirrors/

source:
https://www.netbsd.org/releases/formal-7/NetBSD-7.1.html

Google's 2017 Summer of Code Program

The FreeBSD Project announced that it will participate in Google's 2017 Summer of Code program, which funds summer students to participate in open source projects. This will be the FreeBSD Project's thirteenth year in the program, having mentored over 200 successful students through summer-long coding projects between 2005 and 2016.

Past successful projects have included improvements to Linux ABI emulation, NFSv4 ACLs, TCP regression testing, FUSE file system support, and countless other projects. Many students go on to become FreeBSD developers, as well as participating in FreeBSD developer events around the world through continuing support from the FreeBSD Foundation.

source:
https://www.freebsd.org/projects/summerofcode.html

OpenSSH 7.5 Released

OpenSSH is the premier connectivity tool for remote login with the SSH protocol. It encrypts all traffic to eliminate eavesdropping, connection hijacking, and other attacks. In addition, OpenSSH provides a large suite of secure tunneling capabilities, several authentication

methods, and sophisticated configuration options. The OpenSSH suite consists of the following tools:

Remote operations are done using ssh, scp and sftp.

Key management with ssh-add, ssh-keysign, ssh-keyscan and ssh-keygen.

The service side consists of sshd, sftp-server and ssh-agent.

OpenSSH is developed by a few developers of the OpenBSD Project and made available under a BSD-style license.

source: https://www.openssh.com/

iXsystems Launches FreeNAS Corral, an Open Source Solution for Building Hyper-converged Infrastructures

FreeNAS Corral, the latest version of FreeNAS, combines sophisticated storage, virtualization, containers, and GUI management in a brand new interface
iXsystems, the industry leader in storage and servers driven by Open-Source, released FreeNAS 10 and unveiled the FreeNAS Corral brand. With FreeNAS Corral (formerly FreeNAS 10), iXsystems introduces the next generation of the world's most popular Open-Source software-defined storage software. FreeNAS Corral extends FreeNAS' enterprise-grade storage capabilities by adding a virtual machine and Docker container management. FreeNAS Corral enables the integration of software-defined storage into VMs and provides persistent storage for Docker containers. These enhancements are provided through a re-designed graphical user interface (GUI) and a powerful command line interface (CLI), making FreeNAS both easier to use and more capable than ever. The new Corral name represents what this release does best: corralling data, virtual machines, containers, and storage services under one management interface.

"FreeNAS Corral catapults the world's most popular storage OS into a new category by combining FreeNAS' renowned storage services with Docker containers and full-machine virtualization capabilities. We're now enabling users and developers to build hyper-converged solutions to support their web-scale applications," said Brett Davis, Executive Vice President of iXsystems. "The Corral name represents the next generation in enterprise grade services contributed by iXsystems to the Open Source community. Only a new name could do justice to such a giant evolutionary step."
FreeNAS Corral introduces an intuitive new graphical user interface, a scriptable command line interface, and a powerful websocket API that can automate and control every aspect of the FreeNAS Corral software. It also includes the bhyve hypervisor for virtualization and Docker container services. FreeNAS Corral includes easy to use VM templates, which provide fully set up, pre-installed versions of multiple guest operating systems including TrueOS, FreeBSD, SmartOS, and several GNU/Linux distributions, including CentOS, Debian, and Ubuntu. VMs can also be created for a variety of Windows environments using a user-provided installation media.
Rather than using the cloud to develop, test and deploy applications, FreeNAS Corral, with its storage, VM, and container services, can be easily used instead.

When it comes to storage, early users found provisioning storage with FreeNAS Corral to be more intuitive and accomplished in a shorter amount of time than with previous versions of FreeNAS. These early users also found that FreeNAS Corral's VM and Docker container support enabled them to easily host their application solutions while using FreeNAS Corral's storage services. FreeNAS Corral seamlessly supports Docker containers from dockerhub, enabling DevOps teams to manage, deploy, and scale trusted and business-ready

applications across FreeNAS Corral instances cooperatively with the cloud.
“Thousands of early FreeNAS 10 testers have already seen and deployed many of the revolutionary enhancements that FreeNAS Corral is delivering. This release of FreeNAS Corral gives even more users the opportunity to see all the revolutionary enhancements that continue to make FreeNAS the world’s leading Open-Source storage system. It provides all the features of FreeNAS while remaining 100% Open-Source, also fully leveraging other open-source infrastructures like GitHub and Docker Hub,” said Jordan Hubbard, CTO of iXsystems and head of the FreeNAS Corral project. “We look forward to continued collaboration with the Open Source development community!”
FreeNAS Corral is a ground-up rewrite of FreeNAS that allows for future innovation in the product while supporting all the storage features of FreeNAS 9.10. Users can download FreeNAS Corral at freenas.org/download or upgrade their 9.10.x systems in place by selecting the FreeNAS-Corral-STABLE train from the Update tab of the FreeNAS GUI or installing FreeNAS Corral from the ISO image and picking the option to install into a new Boot Environment.

source:
https://www.ixsystems.com/blog/ixsystems-launches-freenas-corral-open-source-solution-building-hyper-converged-infrastructures/

Qt 5.7.1, KDE Frameworks 5.31 Landed

The KDE-FreeBSD team announced the immediate availability of Qt 5.7.1 and KDE Frameworks 5.31 in the official FreeBSD ports tree; check out FreshPorts for the latest ports news.

This release adds one new KDE Framework, Kirigami 2. It also enables the Qt4 and Qt5 ports to live together, more harmoniously. In particular, it adds misc/qtchooser, which allows developers and sysadmins to manage multiple concurrent Qt installations. We advise users to consult the UPDATING entry for 20170218.

source:
https://freebsd.kde.org/news.php#itemQt571KDEFrameworks531landed

EuroBSDcon 2017

EuroBSDcon is the premier European conference on the open-source BSD operating systems attracting about 250 highly skilled engineering professionals, software developers, computer science students and professors, and users from all over Europe and other parts of the world. The goal of EuroBSDcon is to exchange knowledge about the BSD operating systems, facilitate coordination and cooperation among users and developers.
The conference will be held at the Espace Saint Martin in Paris the 21-24 September 2017

Call for Proposals
The tutorials will be held on Thursday and Friday to enlisted participants and the talks are presented to conference attendees on Saturday and Sunday.
The call for Talk and Presentation proposals period will close on April 30th, 2017.

Call for Talk and Presentation Proposals (CFP)
The EuroBSDcon program committee is inviting BSD developers and users to submit innovative and original talk proposals not previously presented at other European conferences. Topics of interest to the conference include but are not limited to applications, architecture, implementation, performance and security of BSD-based operating systems, as well as topics concerning the economic or organizational aspects of BSD use. Presentations are expected to be 45 minutes and are to be delivered in English.

Call for Tutorial Proposals
The EuroBSDcon program committee is also inviting qualified practitioners in their fields to submit proposals for half or full day tutorials on topics relevant to development, implementation and use of BSD-based systems. Half-day tutorials are expected to be 2.5 to 3 hours and full-day tutorials 5 to 6 hours. The tutorials and talks are to be delivered in English.

Submissions
Proposals should be sent by email to submission at eurobsdcon.org.

Source: https://2017.eurobsdcon.org/news/

Among clouds Performance and Reliability is critical

Download syslog-ng Premium Edition product evaluation here

Attend to a free logging tech webinar here

BalaBit IT Security

www.balabit.com

syslog-ng log server

The world's first High-Speed Reliable Logging™ technology

HIGH-SPEED RELIABLE LOGGING

- above 500 000 messages per second
- zero message loss due to the Reliable Log Transfer Protocol™
- trusted log transfer and storage

Building a PCI Compliant Infrastructure on AWS

Infrastructure security has become one of the most important topics in the IT industry nowadays. More and more we witness attacks being carried out against big and small companies. That is why it is so important to know how to protect your infrastructure and your systems. To help organizations do that, there are a number of different security standards: PCI DSS, Red Flag, HIPAA/HITECH Security, NIST, NERC, ISO 27002 etc. In this article, we will focus on PCI DSS.

PCI DSS is a security standard for companies that deal with credit card information from schemes like MasterCard, Visa, American Express, JCB and Discover. However, even if your company does not handle any credit card information, it would still be a great idea to implement some of PCI's security standards.

Solution

We will be building a quite complex AWS infrastructure and we will be deploying an application called Guestbook. The Guestbook is a program written in Go which adds guests to a list using Redis. This application is part of the sample applications which are provided by Kubernetes for users to try Kubernetes out. The reason why I chose to use this sample application instead of coding one from scratch, is because the main objective of this article is to show you how to protect your infrastructure by applying security standards, not to demonstrate application development.

Everything that you will learn here can be applied to any infrastructure running any application.

Technology stack

We will be using the following operating system, cloud provider and programming language:

- Amazon Web Services
    - VPC (Subnets, Internet Gateway, Route Tables, NAT Gateways)
    - ElastiCache (Redis)
    - EC2
    - Elastic Load Balancer (ELB)
    - Route53
- Ubuntu 16.04 LTS

Go programming language - you do not need to have previous programming experience in Go as the code will be provided

Infrastructure Components

This is the AWS infrastructure that we will be building (see Figure 1):

- An Amazon VPC with six (6) subnets - three (3) public subnets and three (3) private subnets
- A Redis cluster
- An Elastic Load Balancer (ELB) which will be deployed to the Demilitarized Zone (DMZ)
- A Bastion host which will also be deployed to the DMZ
- An EC2 instance which will run our application
- Security Groups to protect the bastion host, the ELB and the instance

- A NAT Gateway to allow the private EC2 instance to communicate with the Internet

Note that the resources above might be deployed to any of the availability zones. For example, the guestbook EC2 instance can be deployed to us-east-1a, us-east-1b or us-east-1c, so long as it is deployed to a private subnet.

Infrastructure Requirements

PCI DSS contains over 300 requirements to be fully compliant. Due to space constraints, we will implement just a small set of these requirements.

By the end of this article, you will have implemented the following PCI requirements (the number in parentheses is the PCI section number):

1 - Is a firewall required and implemented at each Internet connection and between any demilitarized zone (DMZ) and the internal network zone?(1.1.4a)

2 - Are firewall and router rule sets reviewed at least every six months? (1.1.7b)

3 - Do firewall and router configuration standards require review of firewall and router rule sets at least every six months? (1.1.7a)

4 - Is inbound and outbound traffic restricted to that which is necessary for the cardholder data environment? (1.2.1a)

5 - Is all other inbound and outbound traffic specifically denied (for example by using an explicit "deny all" or an implicit deny after allow statement)? (1.2.1b)

6 - Is a DMZ implemented to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports? (1.3.1)

7 - Is outbound traffic from the cardholder data environment to the Internet explicitly authorized? (1.3.4)

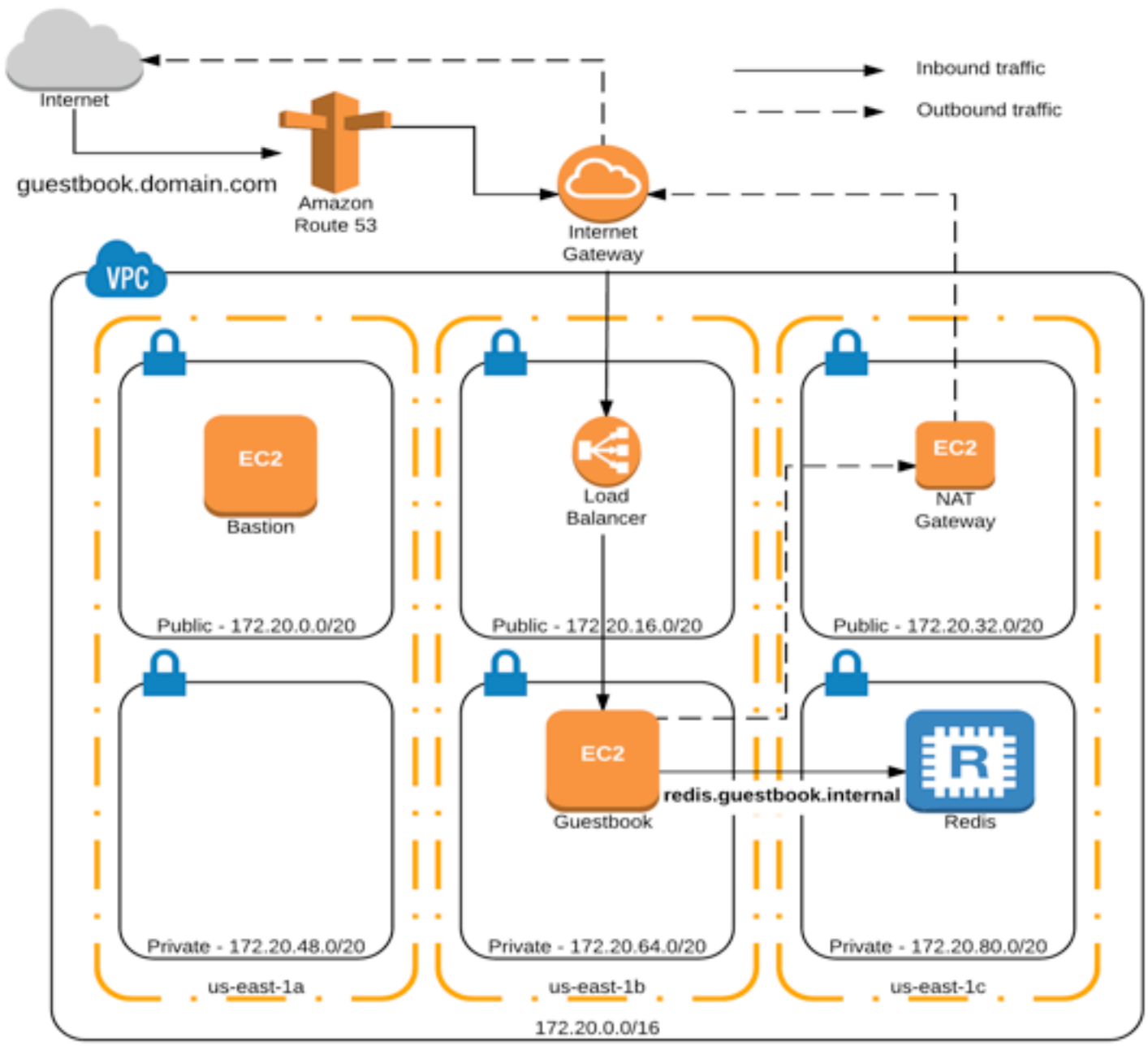


Figure 1. A graphical view of the infrastructure

8 - Are only established connections permitted into the network? (1.3.5)

9 - Are system components that store cardholder data (such as a database) placed in an internal network zone, segregated from the DMZ and other untrusted networks? (1.3.6)

10 - Are configuration standards developed for all system components and are they consistent with industry-accepted system hardening standards? (2.2a)

Sources of industry-accepted system hardening standards may include, but are not limited to, SysAdmin Audit Network Security (SANS) Institute, National Institute of Standards Technology (NIST), International Organization for Standardization (ISO), and Center for Internet Security (CIS).

11 - Is only one primary function implemented per server, to prevent functions that require different security levels from co-existing on the same server? (2.2.1a)

12 - For TLS implementations, is TLS enabled whenever cardholder data is transmitted or received? (4.1e)

13 - Are development/test environments separate from the production environment? (6.4.1a)

14 - Is there a formal Risk Mitigation and Migration Plan in place for all implementations that use SSL and/or early TLS (other than as allowed in A2.1), that includes:(A2.2)

15 - Are methods in place to prevent the disclosure of private IP addresses and routing information to the Internet? (1.3.7a)

Note: Methods to obscure IP addressing may include, but are not limited to:

Network Address Translation (NAT)

Placing servers containing cardholder data behind proxy servers/firewalls

Removal or filtering of route advertisements for private networks that employ registered addressing.

Internal use of RFC1918 address space instead of registered addresses.

As soon as we build a piece of infrastructure which complies to any of the requirements above, there will be a section to discuss how the requirement was complied. And even if we do not comply entirely, we will be discussing what else should be done so the requirement is fully complied.

Ready to rock?

Virtual Private Cloud (VPC) and Subnets

Let’s start by creating a Virtual Private Cloud (VPC) and a few subnets. If you haven’t logged in to the AWS console, do so now. After you log in, head over to the

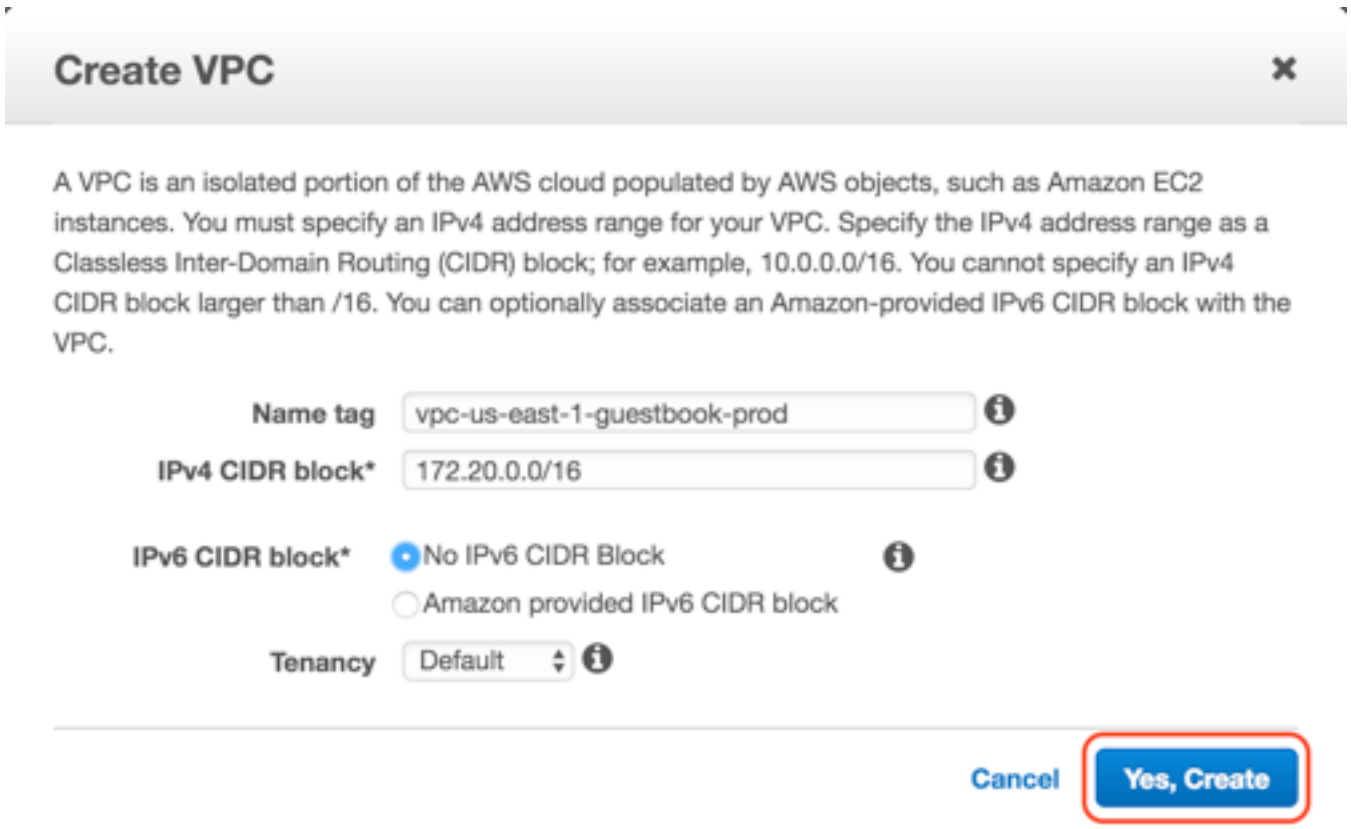


Figure 2. Create VPC

VPC console, click on **Create VPC** (see Figure 2) and use the following information:

For the **Name Tag**, type in *vpc-us-east-1-guestbook-prod* (this is just a suggestion, you can choose any name you would like). For the **IPv4 CIDR block**, enter the Class B address 172.20.0.0/16. Then, select *No IPV6 CIDR Block* and *Default* for **IPv6 CIDR block** and **Tenancy**, respectively.

After your VPC is created, select it and click on **Actions** at the top. Then, click on **Edit DNS Hostnames** and select Yes. You will understand why we are doing this later on.

Now that we defined the CIDR block of our VPC, we need to calculate the address that will be use for the 6 subnets.

The network will have the address 172.20.0.0/16 (default netmask is 255.255.0.0), which means that out of 32 bits, 16 bits are reserved for the network, and 16 bits will be used for subnets and hosts:

1 1 1 1 1 1 1 1 . 1 1 1 1 1 1 1 1 . 0 0 0 0 0 0 0 0 . 0 0 0 0 0 0 0 0

|---------- Network ----------| |------ Subnets and Hosts ------|

The formula to calculate the number of subnets in a network is as follows:

2^x

Where x is the number of 1s that were set for the subnet. Since we need only 6 subnets, then x = 3, or 2^3 = 8 subnets seems to be appropriate at first. However, 8 subnets is quite tight, especially if later we wish to create three (3) or more subnets for some reason. I suggest we go for x = 4, which would give us 2^4 = 16 subnets and a netmask of 255.255.240.0 (/20).

Now, to calculate the number of hosts per subnet, the following formula is used:

2^x - 2

Where x is the number of 1s that were set for hosts. In this formula, it is subtracted two (2) from 2^x because, for each subnet, the first IP address is the address of the subnet and last IP address is the broadcast address (used to send datagrams to all hosts). Since, out of the remaining 16 bits, 4 bits will be used for the subnets, 12 bits will be used for hosts, which gives us 2^12 - 2 = 4094 hosts per subnet

Next step will be to actually calculate the addresses of each of the six (6) subnets.

Here's how we'd calculate the address range of the first subnet:

11111111.11111111.00000000.00000000 (first address)
172 . 20 . 0 . 0

11111111.11111111.00001111.11111111 (last address)
172 . 20 . 15 . 255

So, the first IP address (subnet address) would be 172.20.0.0/20 and the last IP address (broadcast address) would be 172.20.15.255/20. The IP address between these two (2) will be used for hosts.

Let's calculate now the address range for the second subnet:

11111111.11111111.00010000.00000000 (first address)
172 . 20 . 16 . 0

11111111.11111111.00011111.11111111 (last address)
172 . 20 . 31 . 255

The first IP address (subnet address) would be 172.20.16.0/20 and the last IP address (broadcast address) would be 172.20.31.255/20. I believe you already noticed a pattern which will help us calculate the rest of the address ranges without writing down 0s and 1s. The first and second octet (8 bits) never change. The third octet increases 16 units for every subnet, and the fourth octet is always 0 (for the first address) and 255 (for the last address. We can, therefore, conclude from that, that we need to add 16 units to the third octet for each subnet. This will give us the following address ranges:

First subnet: 172.20.0.0/20 - 172.20.15.255/20

Second subnet: 172.20.16.0/20 - 172.20.31.255/20

Third subnet: 172.20.32.0/20 - 172.20.47.255/20

Fourth subnet: 172.20.48.0/20 - 172.20.63.255/20

Fifth subnet: 172.20.64.0/20 - 172.20.79.255/20

Sixth subnet: 172.20.80.0/20 - 172.20.95.255/20

Phew! We finally have all address ranges and can create the subnets. However, before we create the subnets, let's

snpr-us-east-1a-prod	subnet-4fa8b606	available	vpc-1a85fe7c | ...	172.20.48.0/20	4091	us-east-1a
snpr-us-east-1b-prod	subnet-bf290de4	available	vpc-1a85fe7c | ...	172.20.64.0/20	4091	us-east-1b
snpr-us-east-1c-prod	subnet-e071e285	available	vpc-1a85fe7c | ...	172.20.80.0/20	4091	us-east-1c
snpu-us-east-1a-prod	subnet-80abb5c9	available	vpc-1a85fe7c | ...	172.20.0.0/20	4091	us-east-1a
snpu-us-east-1b-prod	subnet-f8290da3	available	vpc-1a85fe7c | ...	172.20.16.0/20	4091	us-east-1b
snpu-us-east-1c-prod	subnet-2271e247	available	vpc-1a85fe7c | ...	172.20.32.0/20	4091	us-east-1c

Figure 3. Subnet's List

quickly discuss a naming convention. We will use the following naming convention for naming the subnets:

```
sn[pu/pr]-[region][availability zone]-[environment]
```

Note: *pu* will be used for public subnets and *pr* will be used for private subnets.

For instance, if we wish to create a public subnet for the production environment on the North Virginia region and availability zone A, we would name the subnet:

```
snpu-us-east-1a-prod
```

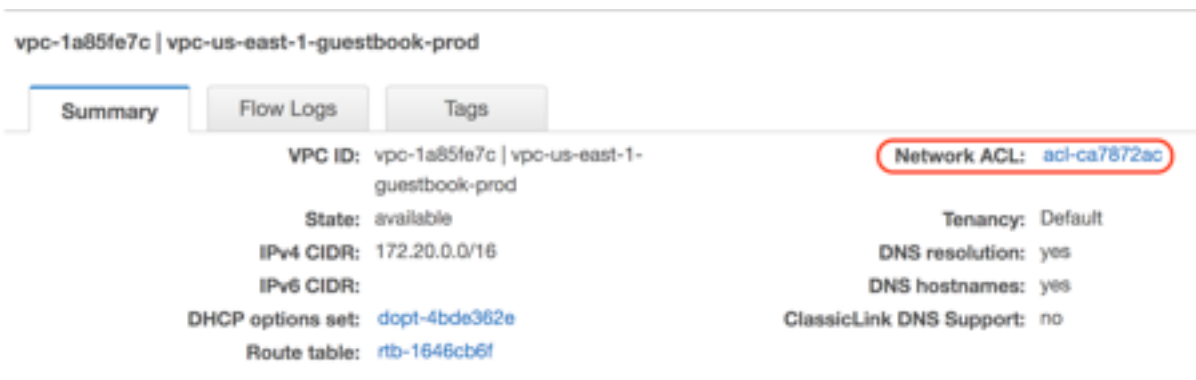


Figure 4. Information about this VPC, including the NACL's ID

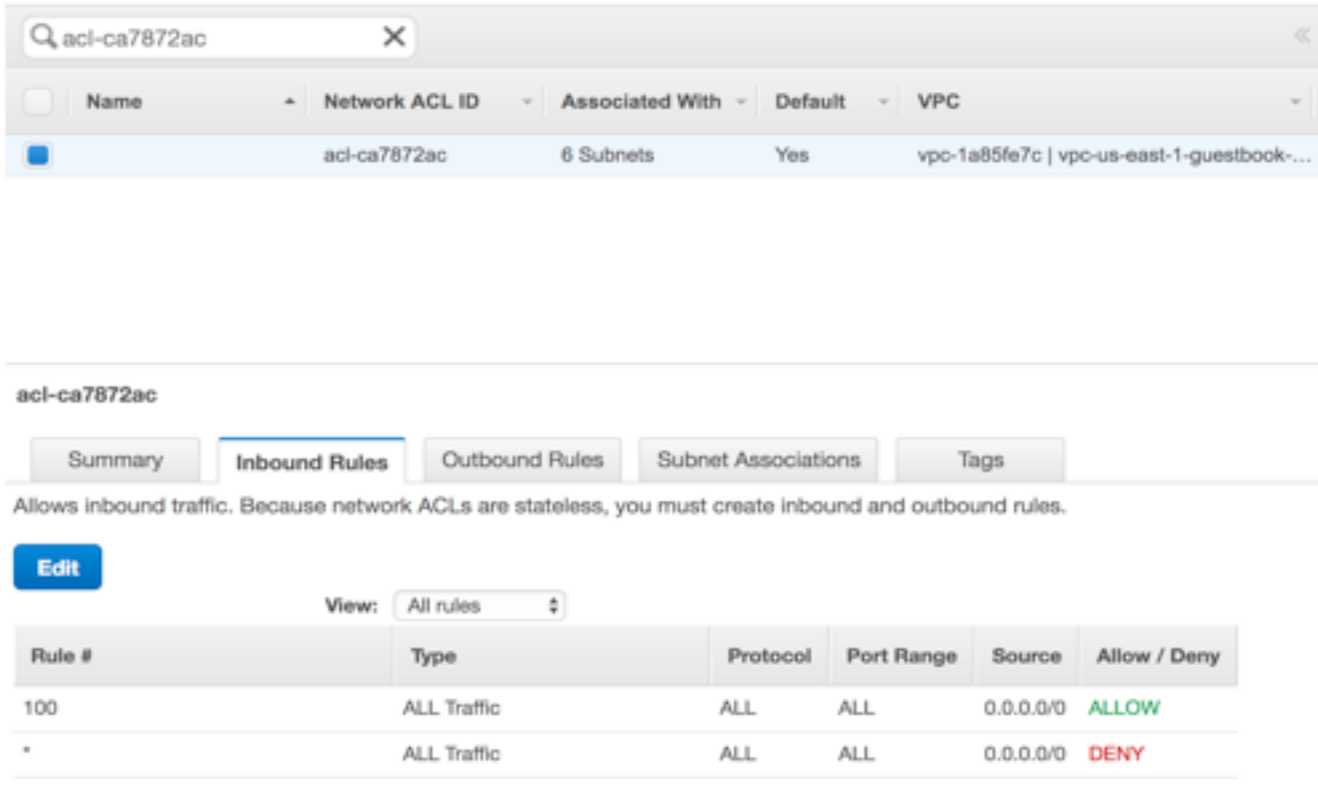


Figure 5. **Inbound Rules** tab

On the left panel of the VPC console, click on **Subnets** and create six (6) subnets using the information below:

Name tag: *snpu-us-east-1a-prod* **VPC:** *vpc-us-east-1-guestbook-prod* **Availability Zone:** *a* **IPv4 CIDR block:** *172.20.0.0/20*
Name tag: *snpu-us-east-1b-prod* **VPC:** *vpc-us-east-1-guestbook-prod* **Availability Zone:** *b* **IPv4 CIDR block:** *172.20.16.0/20*
Name tag: *snpu-us-east-1c-prod* **VPC:** *vpc-us-east-1-guestbook-prod* **Availability Zone:** *c* **IPv4 CIDR block:** *172.20.32.0/20*
Name tag: *snpr-us-east-1a-prod* **VPC:** *vpc-us-east-1-guestbook-prod* **Availability Zone:** *a* **IPv4 CIDR block:** *172.20.48.0/20*
Name tag: *snpr-us-east-1b-prod* **VPC:** *vpc-us-east-1-guestbook-prod* **Availability Zone:** *b* **IPv4 CIDR block:** *172.20.64.0/20*
Name tag: *snpr-us-east-1c-prod* **VPC:** *vpc-us-east-1-guestbook-prod* **Availability Zone:** *c* **IPv4 CIDR block:** *172.20.80.0/20*

After creating all the subnets, the console should list as it is shown on Figure 3.

Before we move on to create the Internet Gateway, we need to change a rule in the VPC's Network Access Control List (NACL). Briefly, a NACL is a firewall at the subnet level. When we created the VPC, a rule that allows all kinds of traffic from everywhere was created as well in the default NACL. What we need to do is restrict this rule to only allow TCP traffic. Click on **Your VPCs** and select the VPC you have just created. At the bottom of the page, there will be some information about this VPC, including the NACL's ID (See Figure 4).

Click on the Network ACL ID. Then, select this ACL and click on the **Inbound Rules** tab (see Figure 5).

Then, click on **Edit**. For the **Type** column (do not create a new rule, just edit the existing one), select **ALL TCP** and click on **Save**. You should have the following rules as it is shown on Figure 6.

Now, click on the **Outbound Rules** tab and do exactly the same. After you change the rule, you should have the following as it is shown on Figure 7.

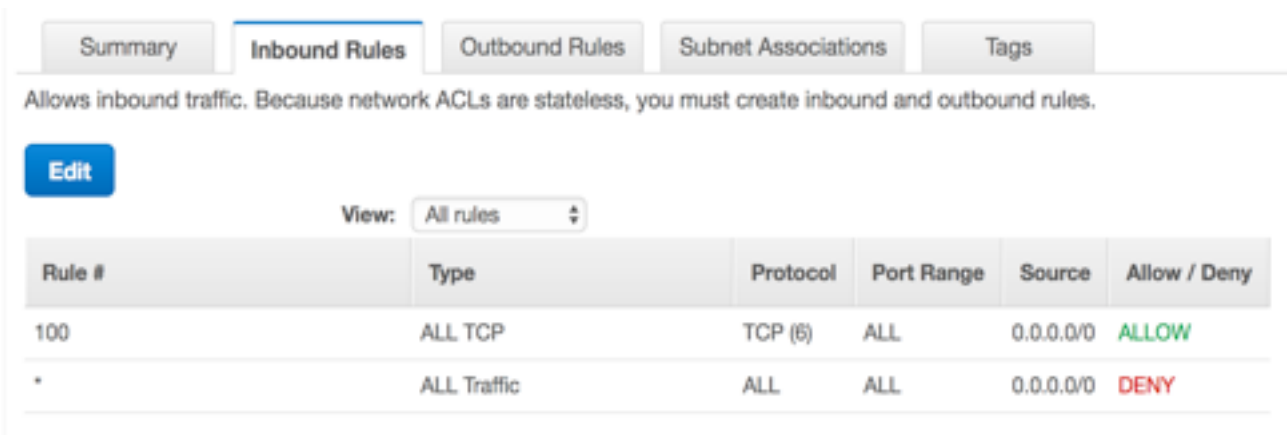


Figure 6. Rules

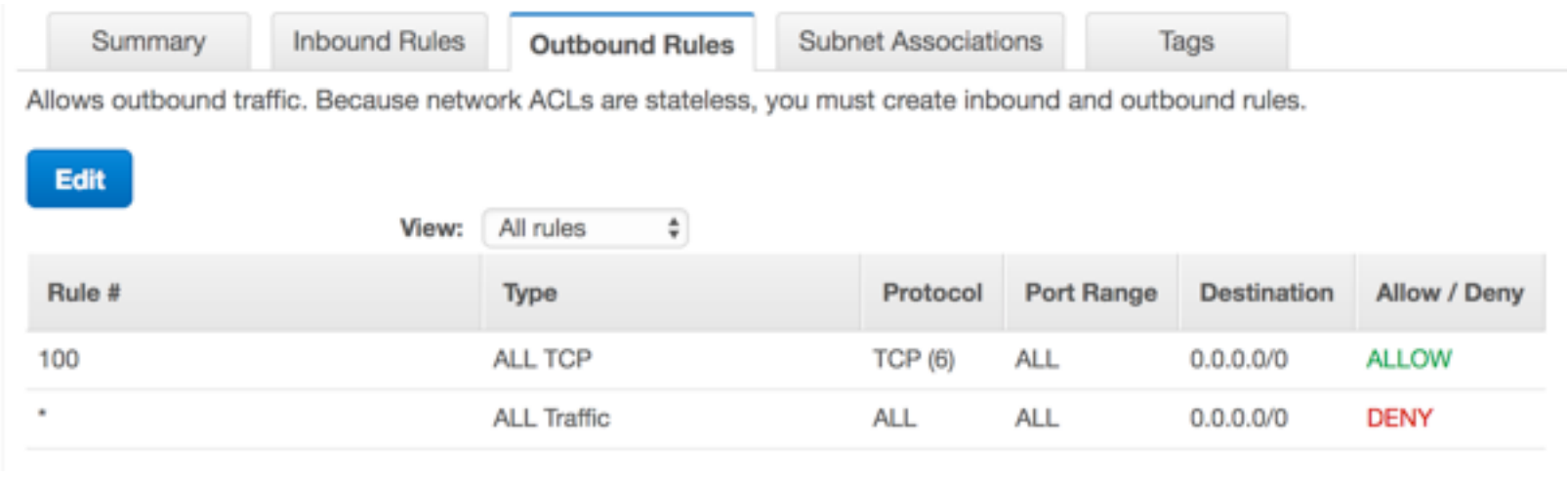


Figure 7. Changing the rule

Compliance achieved: requirements 1.2.1b, 1.3.1

By restricting inbound and outbound traffic, we achieve the following requirement:

Is all other inbound and outbound traffic specifically denied (for example by using an explicit "deny all" or an implicit deny after allow statement)? (1.2.1b)

If you take a look at both inbound and outbound rules, you will see that the last one (which is default for all Network ACLs) is a deny-all rule.

Also, because we created public subnets (Demilitarized Zone), our infrastructure complies with the following requirement:

Is a DMZ implemented to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports? (1.3.1)

There is one more requirement that we haven't complied yet, but it would be really easy to comply with:

Are development/test environments separate from the production environment? (6.4.1a)

If we wanted to have a few more environments (development, staging, testing etc) instead of just production, we'd create new VPCs and follow the same setup that we will do throughout the rest of the article - and that would be enough to make our infrastructure comply with requirement 6.4.1a.

Internet Gateway (IGW)

In order for users to interact with our application, we need a mechanism that will allow our server to communicate with the Internet. This mechanism is a VPC component called Internet Gateway.

If you are still on the VPC management console, look for **Internet Gateways** on the left-hand panel. Then, hit **Create Internet Gateway** (see Figure 8).

Name your Internet Gateway *igw-eu-east-1-guestbook* and click on **Yes, Create**. After you create it, you will notice that in the **state** column it reads **detached**. That is because an Internet Gateway needs to manually be attached to a VPC. Select your newly created IGW, then click on **Attach to VPC**. Select your VPC and hit **Yes, Attach**. Well done! Next, we will need a Network Translation Address (NAT) instance.

Network Translation Address (NAT) instances

When you have instances running on private subnets, there needs to be a way for them to communicate with the Internet. And I bet you might have just thought: "Isn't that what the Internet Gateway is for?" - You're right, it is! However, instances running on private subnets do not have a public IP address, which means that their traffic would not be routable through the Internet. This means that these private instances need to "borrow" a public IP address from some other machine. And this other machine, on the AWS context, is called Network Translation Address (NAT). Basically, you would run NAT instances on the public subnets and would give them a public IP address. Then, you would configure the private subnets' route table to send all traffic coming from the private instances to the NAT. The NAT, in turn, would forward this traffic to the Internet using its own public IP address, making the traffic routable. When the traffic comes back, the NAT forwards it back to the instance.

Note that this whole explanation is for the case when the **instance** initiates the traffic. If the user on the Internet initiates the traffic, however, it would not go through the NAT because the user wouldn't know the private IP

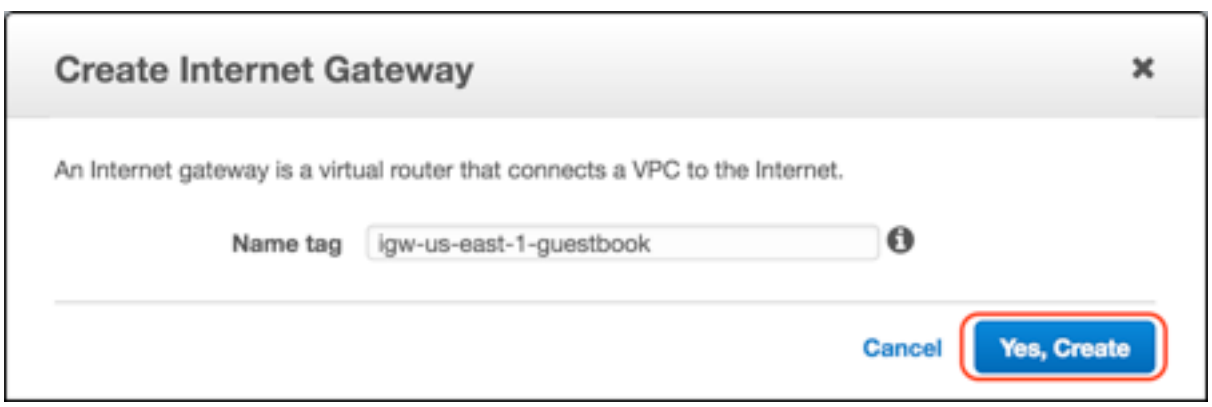


Figure 8. Create Internet Gateway

address of the instance. In this case, we would need an Elastic Load Balancer, which would act as a proxy (we'll get there soon).

There are two main ways to create NAT instances. The first and hardest way, would be to launch EC2 instances on the public subnets, using a NAT AMI. If you click on **Launch Instance**, then click on **Community AMIs** (on the left-hand side) and type in "NAT", you will have plenty of NAT AMIs to choose from. Also, another important thing that you would need to configure is the **Source/Destination Check**. Remember when I said that the NAT would forward a private instance's traffic to the Internet, and from the Internet back to the instance? Well, since the NAT instance is neither the source nor destination of the traffic, it needs to be configured to ignore the source and destination information on the packets. If you are still confused, don't worry because it is quite complex. But, let me give you more examples. When an EC2 instance receives network traffic, it checks on the packets whether the instance itself is the source or destination of the traffic. If the instance is neither the source nor the destination of the traffic, it will drop the packet, thus ignoring the traffic. Now, if the NAT instance were to do that, all the traffic coming from the private instances would be dropped, because the NAT, in most cases, is neither the source nor the destination of the traffic. This is why NAT instances need to be configured not to check the source/destination information on the packets. Ok, let's talk about now the second way to create NAT instances (which is far easier). The second way would be to use Amazon's NAT service called **NAT Gateways**. NAT Gateways are NAT instances which are managed by Amazon and can support bursts up to 10 Gbps of bandwidth. This is quite handy because if you launch your own EC2 instance and there's a huge burst of bandwidth, your NAT instance might not be able to handle the whole traffic depending on its size.

For this article, we will use NAT Gateways because of its convenience. If you wish to use NAT instances, however, you can follow this tutorial from AWS: http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_NAT_Instance.html

Before you create a NAT Gateway, you will need one piece of information: the ID of subnet to which you would like AWS to deploy a NAT Gateway. On the VPC management console, click on Subnets and copy the ID of one of the ***public*** subnets. Then, click on **NAT Gateways** on the left-hand side, and click on **Create NAT Gateway** (see Figure 9).

Paste the subnet ID on the first text field. Now, the second text field asks for an Elastic IP Allocation ID. An Elastic IP on AWS is just a public IP address which will be allocated to your instances. If you already have created Elastic IPs, grab the allocation ID from one of them and paste here, otherwise, click on **Create New EIP**. If you hit **Create New EIP**, AWS will create an Elastic IP automatically and fill in the text field for you. When you're done, click on **Create a NAT Gateway**.

For this article we will only create one NAT Gateway as NAT Gateways aren't as cheap as t2.micro instances. In Northern Virginia, for example, a NAT Gateway costs around $32 per month, while a t2.micro instance costs less than $10. If you would like to have a scalable infrastructure, however, you would need to deploy one NAT Gateway per public subnet, so the load is distributed.

Compliance Achieved: requirement 1.3.7a

By creating a NAT instance on the public subnet, we achieve the following requirement:

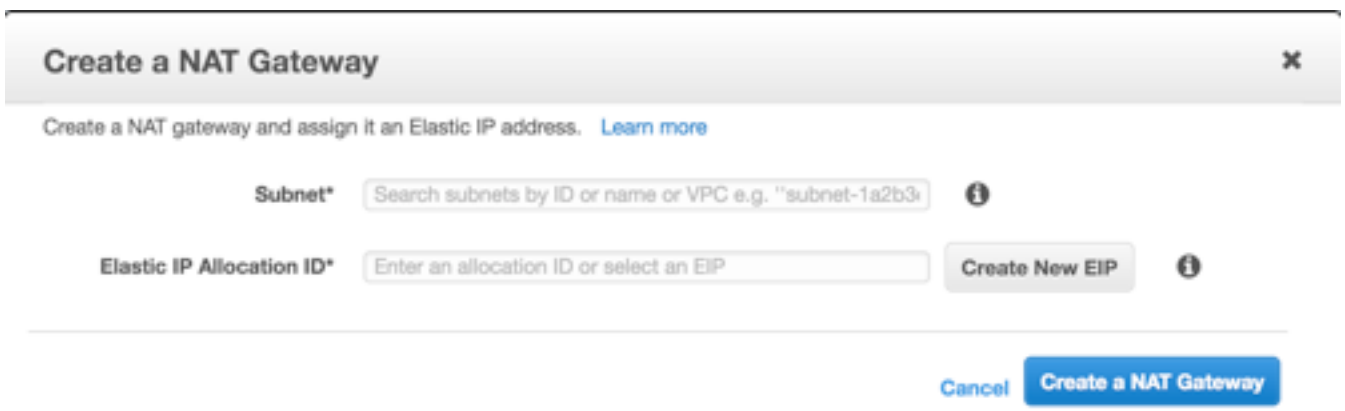


Figure 9. NAT Gateway

19 - Are methods in place to prevent the disclosure of private IP addresses and routing information to the Internet? (1.3.7a)

Note: Methods to obscure IP addressing may include, but are not limited to:

Network Address Translation (NAT)

Placing servers containing cardholder data behind proxy servers/firewalls

Removal or filtering of route advertisements for private networks that employ registered addressing.

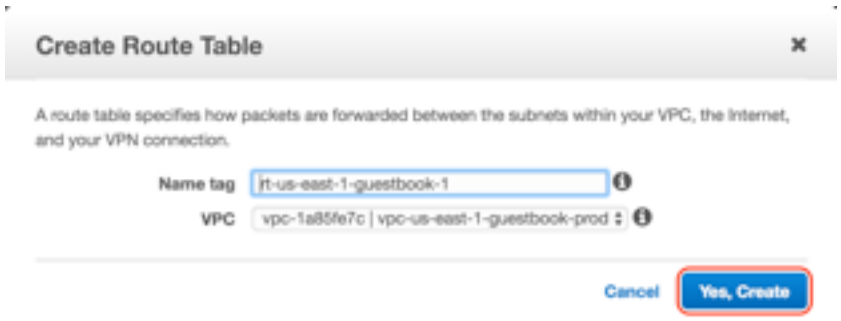
Internal use of RFC1918 address space instead of registered addresses.

Having a NAT instance is just one of the ways to achieve this requirement. We will soon deploy our application behind a proxy, which will also comply with requirement 1.3.7a.

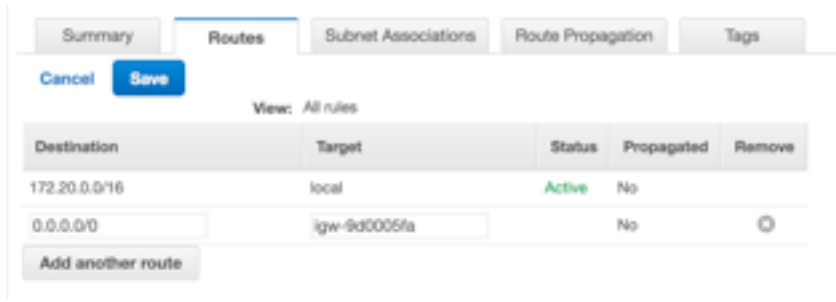
Route Tables

When you create a new VPC, traffic is only routable locally. This means that, even though you create a NAT Gateway and an Internet Gateway, the traffic still wouldn't leave the VPC. In order for that to happen, we need to add a few rules to the route tables. If you click on **Route Tables** (VPC console), you will notice that AWS has already created a default route table for your VPC. In my projects, I always create new route tables and leave the default one unused. However, if you'd like to use the default route table, go ahead. For this article we will create new route tables.

Click on **Create Route Table**, name it *rt-us-east-1-guestbook-1* and select your VPC:

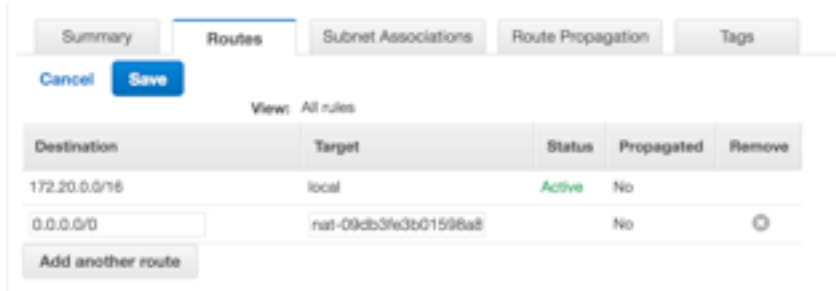


After you hit **Yes, Create**, click on the **Routes** tab at the bottom of the page. This will be the route table of the public subnets, which means that there needs to be a rule which directs the traffic to the Internet Gateway.



Click on **Edit** and then click on **Add another route**. Use *0.0.0.0/0* (which means **any** IP address) as destination and the Internet Gateway's ID as target. This rule is basically saying: "whenever there is a traffic whose destination is not from the 172.20.0.0/16 address space, direct it to the Internet Gateway". Hit **Save** to finish editing.

Now create another route table called *rt-us-east-1-guestbook-2*, add the NAT Gateway as Target and hit **Save**:



To make sure the route table contains valid routes, make sure the **Status** of all entries is **Active**.

After you created both route tables, it is time to associate these with the subnets. Click on **Subnets** and filter by your VPC if necessary. Once you located all the six subnets, for each of them, do the following:

Select the subnet, select the **Route Table** tab and click on **Edit**

If the subnet is **public**, select the *rt-us-east-1-guestbook-1* route table and save

If the subnet is **private**, select the *rt-us-east-1-guestbook-2* route table and save

Security Groups

Security groups are a crucial part of any AWS infrastructure because it has the same functionality as a firewall. However, you can only specify allow rules, and not deny rules. For example, suppose you wanted to deny access to your application to a certain public IP address.

This would not be possible with security groups because you wouldn't be able to create a rule which denies access to the instance to this IP address. The solution to this problem would be to create a Network Access Control List (NACL), instead. Right, so what's the difference between Security Groups and NACL? Here are the main differences:

NACL are applicable at the subnet level, while Security Groups are applicable at the instance level

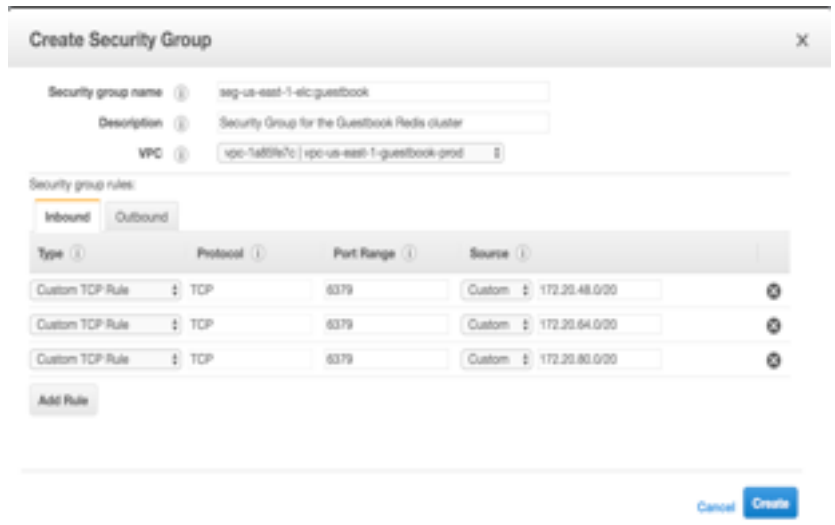
As already mentioned, only allow rules can be specified in Security Groups. While in NACL, you can specify both allow and deny rules;

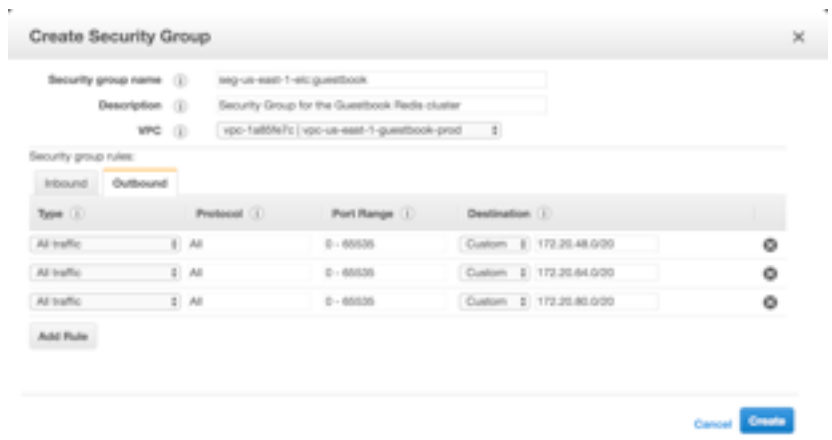
Security Groups are stateful, while NACL are stateless. To understand what this means, you need to understand what's the difference between inbound and outbound rules. When a client initiates the connection with a server, the traffic is analysed by the inbound rules. However, when the server replies to the client, the outgoing traffic is analysed by the outbound rules. Now, think of the following scenario: a client machine wants to access your server on port 5000. If you are using Security Groups, you only need to create an inbound rule allowing traffic on port 5000. However, you **do not** need to create an outbound rule allow outgoing traffic on port 5000 because the Security Group knows that the traffic was initiated by the client and **not** the server. The contrary is also valid - if your server initiates the traffic, you only need to specify an outbound rule allowing traffic on port 5000 (when the traffic comes back from the client, the inbound rules would not stop the traffic because the server initiated it). Now, the same does not happen with NACL. NACLs do not know who started the traffic (they do not keep state), which means that if a connection comes in for the server on port 5000, you need to have an inbound rule to allow the traffic in, and also an outbound rule (**same port**) to allow the traffic back. In summary, Security Groups keep track of the "state" of the traffic, while NACLs do not;

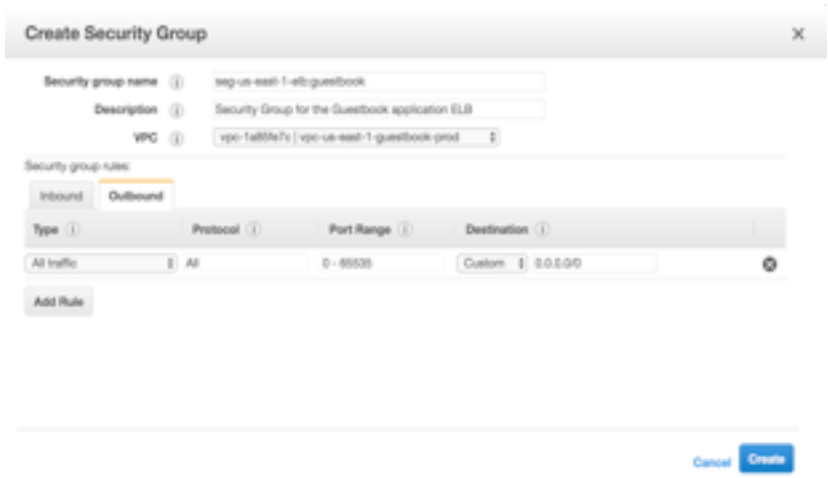
NACLs apply the rules to incoming and outgoing traffic in a certain order. Let's suppose you create a rule numbered 100 which allows UDP traffic on port 4638. Then, you create a rule numbered 200 which denies UDP traffic on the same port. When the traffic comes, the rule number 100 will be applied first, which means that the traffic will be allowed to go through, even though rule number 200 denies the traffic. The same does not happen with Security Groups. Security Groups apply **all** the rules before deciding whether to allow traffic or not, which means that if there is a rule which allows UDP traffic on a certain port, and there is another rule which denies UDP traffic on the same port, the traffic will be denied.

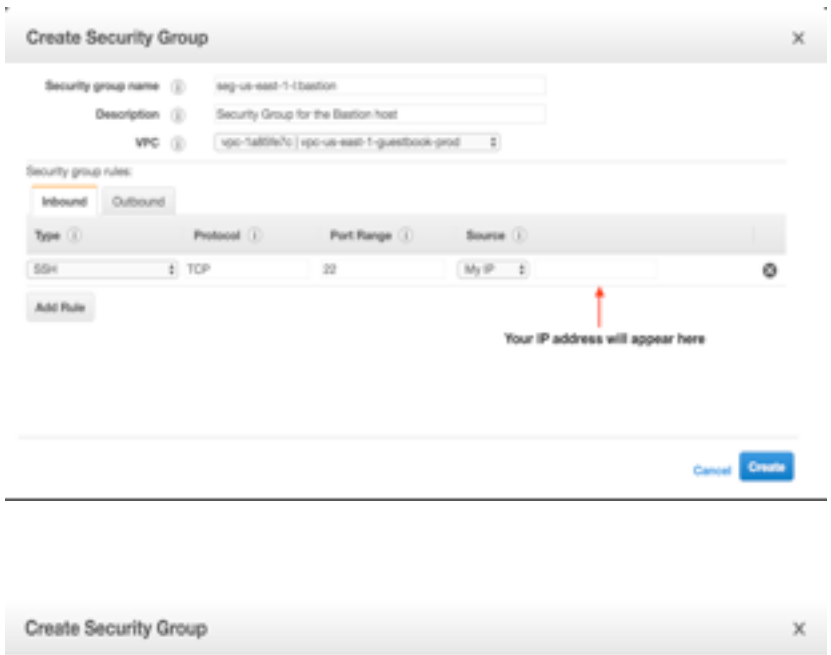
Now that the difference between Network Access Control Lists and Security Groups are clearer, we can move on.

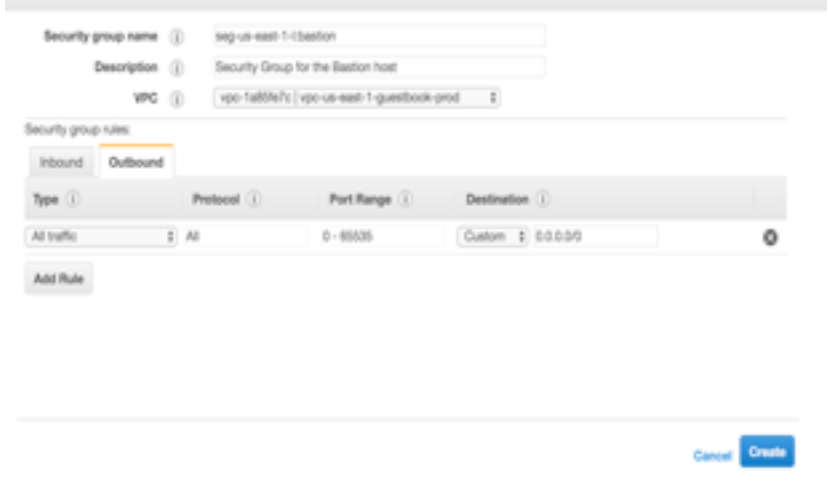
We will be creating three security groups: one for a Redis cluster, one for an Elastic Load Balancer, and one for the sever. Go to the EC2 dashboard and click on Security Groups on the left panel. Create three security groups according to the images below (each security group will have two images - one showing the inbound rules and another showing the outbound rules):

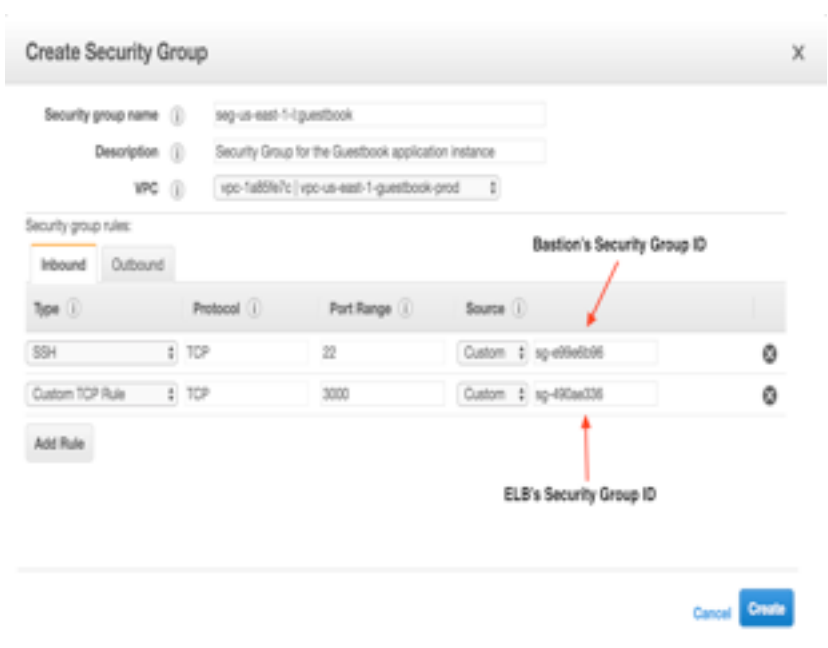












Next, we will create a Redis cluster.

Compliance achieved: requirements 1.1.4a, 1.1.7a, 1.1.7b, 1.2.1a, 1.3.4, 1.3.5

Security Groups help our infrastructure to comply with quite a lot of requirements. Let's discuss one by one.

Is a firewall required and implemented at each Internet connection and between any demilitarized zone (DMZ) and the internal network zone?(1.1.4a)

Our infrastructure complies with requirement 1.1.4a because we have security groups for the Load Balancer and Bastion host (which sit on the DMZ) and a security group for the guestbook instance - allowing only certain traffic from the DMZ.

Are firewall and router rule sets reviewed at least every six months? (1.1.7b)

Do firewall and router configuration standards require review of firewall and router rule sets at least every six months? (1.1.7a)

As long as you write a policy document saying that you are reviewing the security groups rules every six month, you will comply with both of these requirements.

Is inbound and outbound traffic restricted to that which is necessary for the cardholder data environment? (1.2.1a)

Our security groups only allow necessary traffic into our VPC. For instance, only our IP address is allowed when connecting to the bastion host.

Is outbound traffic from the cardholder data environment to the Internet explicitly authorized? (1.3.4)

For each infrastructure component (Redis, load balancer, bastion etc) there is a security group which explicitly authorizes outbound traffic.

Are only established connections permitted into the network? (1.3.5)

Examine firewall and router configurations to verify that the firewall performs stateful inspection (dynamic packet filtering).

As previously explained, Security Groups are stateful (http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html).

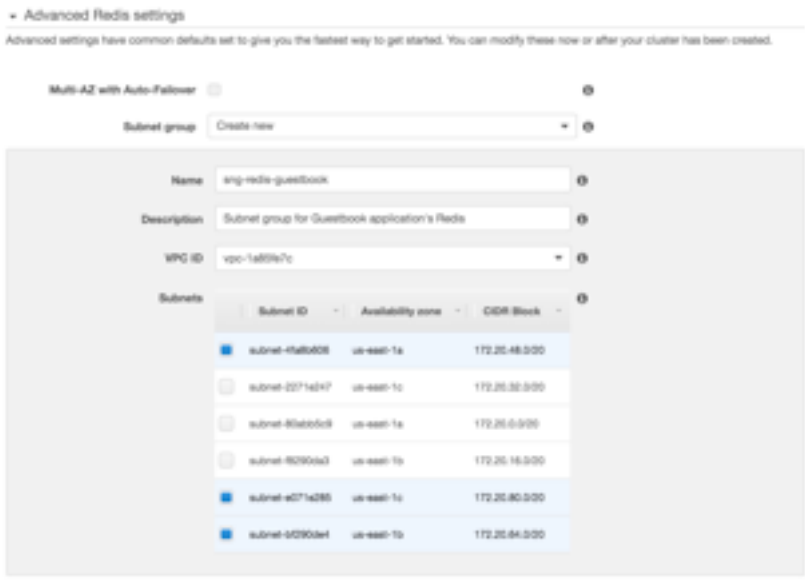
ElastiCache (Redis) Cluster

Like NAT Gateways, AWS also offers a service for Redis, called ElastiCache. ElastiCache can also be used to deploy Memcached clusters. But for this article we will use Redis.

Go to the ElastiCache dashboard, click on Redis on the left panel and hit **Create**:

Name your cluster *redis-guestbook*. You do not need to change **Engine version compatibility**, **Port** or **Parameter group**. Now, in regards to the **Node type**, if this was an application which needed a powerful redis cluster, you would probably go for a R3 node type and at the very least, 1 replica. But since we are building this network for learning purposes, we will stick to the cheapest node type (t2.micro) and no replica.

After you finish filling in all these fields, click on **Advanced Redis settings**. Unselect **Multi-AZ with Auto-Failover** and create a new subnet group if you haven't created one yet:

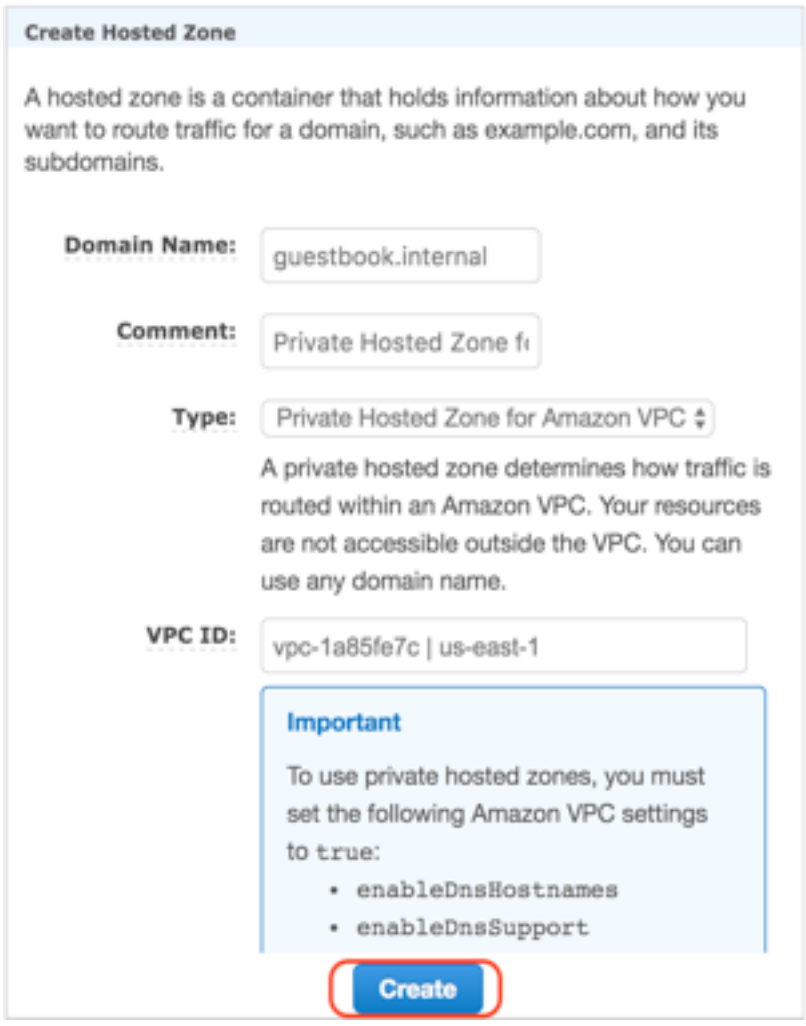


Name your subnet group *sng-redis-guestbook* and add a description. Then, select your VPC and its three **private** subnets. Down below, in Security groups, choose the *seg-us-east-1-elc:guestbook* security group that we have just created.

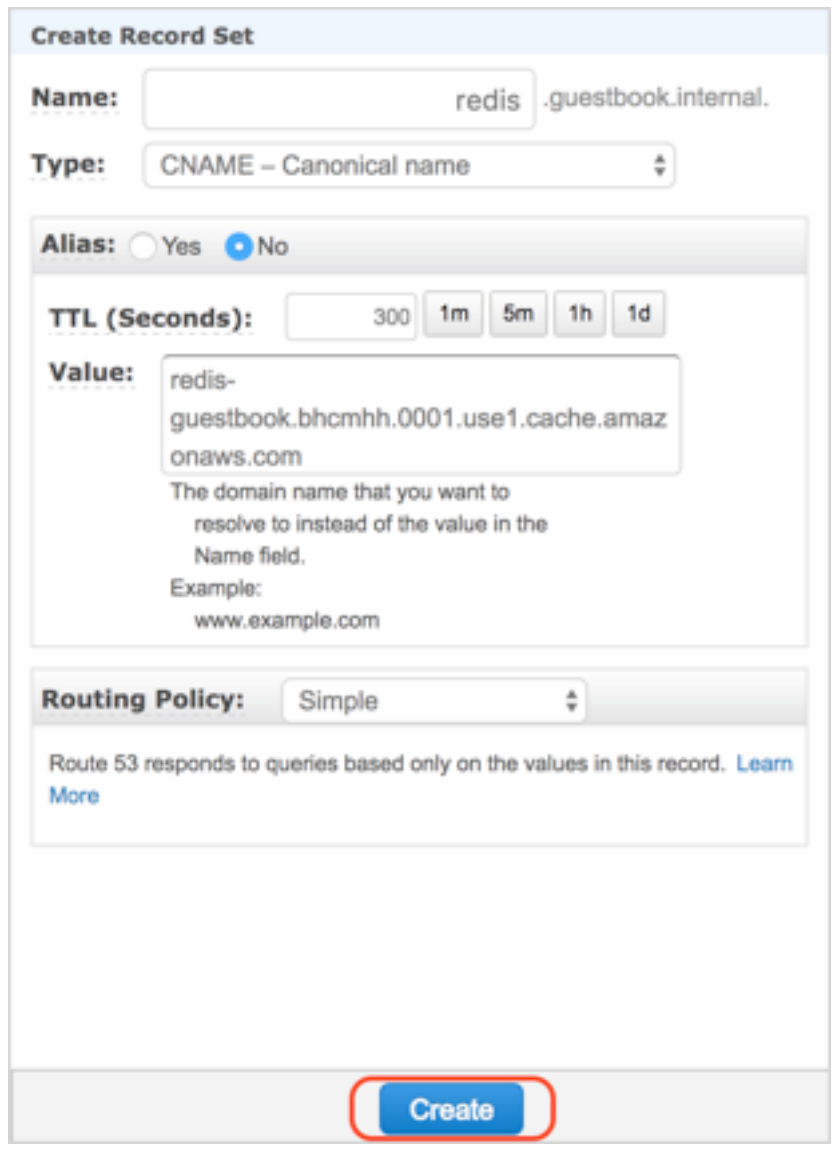
The rest of the configuration you can leave it as is. Finally, click on **Create**.

Before we move on to create the guestbook application's server, we need to configure one more thing. Since each one of us will have a different Redis cluster URLs, we need to agree on the same URL so we don't have to modify the guestbook application. In fact, this is one of the best approaches when it comes to deploying applications to different environments and using different URLs for each environment. We are going to create a private Hosted Zone.

Go to the Route53 Management Console, and click on **Hosted zones**. Then, click on **Create Hosted Zone**. Type in *guestbook.internal* for the **Domain Name** and select the **Type Private Hosted Zone for Amazon VPC**. The console will ask you for a VPC, so select your VPC. Although optional, you can type in a description of this private hosted zone in the **Comment** field (e.g. “Private Hosted Zone for the Guestbook application”). Then click on **Create:**



After the hosted zone is created, we will create a Record Set. The **Name** of this Record Set will be *redis.guestbook.internal*. As for the **Type**, choose **CNAME - Canonical name**. Then, open another tab, go to the ElastiCache dashboard and copy your Redis cluster URL (by clicking on the cluster name and copying the Node’s URL). Paste this URL in the **Value** field. Above the **Value** field, you will see something called **TTL - Time To Live**. The **TTL** defines how long your DNS record will be cached by DNS servers around the world. Suppose you set the TTL to 5 minutes. This means that if you change the DNS record, the DNS servers around the world would show the old value for up to 5 minutes - after the time expires, naturally, the DNS servers will refresh the record. It’s also worth mentioning that if you have a high TTL for a certain DNS record and you wish to change it, prior to actually making the change, lower the TTL so the DNS servers refresh their cache as soon as possible. For instance, if you have a record which has TTL of 86400 seconds (24 hours), change this TTL to something like 600 seconds at least 24 hours before you change the record because this will make your old record to be cached for only 10 min on the DNS servers and your new record will be spread faster. Now that you know what TTL means, leave it as 300 seconds. Click on **Create**:



Remember in the beginning when you created the VPC and then set **Edit DNS Hostnames** to **Yes**? If this option is set to No, private DNS records will not be resolved inside the VPC. By setting it to **Yes** you are telling Amazon that you want private DNS records to be resolved inside your VPC.

Compliance Achieved: requirement 1.3.6

Are system components that store cardholder data (such as a database) placed in an internal network zone, segregated from the DMZ and other untrusted networks? (1.3.6)

Although we are not using Redis to store any cardholder data, we'd still be complying with this requirement as Redis has been deployed to an internal network zone - segregated from the DMZ.

Guestbook Application (EC2)

We will now deploy the guestbook application. Go to the EC2 dashboard and click on **Launch Instance**. Select the **Ubuntu Server 16.04 LTS (HVM), SSD Volume Type** AMI. Then, select t2.micro as Instance Type. Next, on the Configure Instance Details page, select your VPC and select any of the private subnets (any availability zone will do because all of the subnets are configured to send traffic to the NAT instance). You do not need to change any of the remaining fields. Click on **Next: Add Storage**, leave the size of the disk as is (8 GB) and hit **Next: Add Tags**. For the *Name* tag, type in *production - Guestbook* and hit **Next: Configure Security Group**. On the Security Group page, click on **Select an existing security group,** select *seg-us-east-1-l:guestbook* and click on **Review and Launch**. Quickly review your Instance Launch details and hit **Launch**. When asked for an SSH key, select any SSH key that you may already have created, or create a new one.

At this moment you will not be able to connect to this EC2 instance because it was deployed to a private subnet. So what we need to do is to launch a Bastion host on the Demilitarized Zone (DMZ) - a.k.a public subnet.

Bastion Host

A bastion host is an instance that sits on the public subnet and is the gateway through which one can connect to instances on the private subnet.

Launch another EC2 instance like we just did, but change some of the configuration:

- Choose **t2.nano** for the instance size
- Choose a **public** subnet this time
- Use *production - Bastion* as the Name tag
- Select the *seg-us-east-1-l:bastion* security group

After you launch the bastion host, you will notice that there's no public IP address assigned to it. That's because when you create your own VPC, the default behaviour is that instance will not be assigned a public IP address. To fix that, click on **Elastic IPs** on the left panel of the EC2 dashboard, then click on **Allocate New Address**. AWS has changed the interface to create Elastic IPs over the years, but if they give the option to create an Elastic IP and assign it to the bastion host, do so. Otherwise, just allocate a new IP address, then select it, click on **Actions,** click on **Associate Address**, associate it with the Bastion host and you're good to go! Also, it's worth mentioning that the Elastic IP's scope needs to be **VPC**.

Now that we have a bastion host and the guestbook application instance, it's time to connect to the latter. Instead of SSH into the bastion, and then SSH into the guestbook instance, we'll do something different. Add the following snippet of code to your SSH config file (usually located at ~/.ssh/config - if you don't have this file, create one):

```
Host bastion
     HostName <your-Bastion-Elastic-IP>
     User ubuntu
     IdentityFile <path-to-your-private-
key>
     ForwardAgent yes

Host guestbook
     HostName <your-guestbook-instance-
private-IP>
     User ubuntu
     IdentityFile <path-to-your-private-
key>
     ProxyCommand ssh -q -W %h:%p bastion
```

Before saving your SSH config file, add the missing information as described in angle brackets. After that, you only need to do:

```
$ ssh guestboook
```

Piece of cake, right?

If you cannot connect to the guestbook instance, make sure you have the following in place:

- Each public subnet has a route table associated with it where the destination 0.0.0.0/0 has the Internet Gateway as Target
- Each private subnet has a route table associated with it where the destination 0.0.0.0/0 has the NAT instance's ID as Target

- The NAT instance has an Elastic IP associated with it (and if you launched your own instance you have disabled Source/Destination Check)
- The bastion host allows inbound TCP traffic on port 22 from your IP address, and allows outbound TCP traffic to any destination and any port
- The bastion host has been assigned a public Elastic IP address
- The guestbook instance allows inbound TCP traffic on port 22 from the bastion's security group

Deploying the Guestbook application

It is finally time to deploy our application. Log in to the guestbook EC2 instance and type in the following command:

```
$ git clone https://github.com/renansdias/guestbook
```

If git is not installed, use the following commands to install it.

```
$ sudo apt-get update && sudo apt-get install -y git
```

After cloning the guestbook repository, cd into it and list the files. You will see that there's a file called **setup**. This file is a bash script which installs the entire Golang environment and a tool called supervisor (we will use supervisor to keep our Go application running in the background). Just run the setup script and the guestbook application will be running in a few seconds:

```
$ ./setup
```

After the script finishes running, your guestbook application will already be running! To confirm that is the case, type in the following command:

```
$ curl localhost:3000/health
Ok
```

When you make a curl request to the *health* endpoint, the guestbook application should reply "Ok" and the status code of the HTTP request will be 200.

Also, we need to double-check that our application is able to communicate with Redis. Type in the following command:

```
$ curl localhost:3000/info
```

The command above will output some information about your Redis cluster. If you didn't receive any errors and the HTTP status of the request above is 200, then you are good to go!

Elastic Load Balancer (ELB)

We will now create an Elastic Load Balancer to forward traffic to our server. On the EC2 dashboard, click on Load Balancers, then click on **Create Load Balancer**. You will have two options: Application Load Balancer and Classic Load Balancer. Choose the Classic Load Balancer. Now, name your Load Balancer *elb-prod-guestbook*, select the Guestbook's VPC and create two listeners for your load balancer (see Figure 10).

Notice that we are allowing traffic on port 80. The idea would be that all traffic that comes on port 80 is redirected to port 443 so your website enforces traffic over HTTPS. In this article, we will not do that, but that

Figure 10. Name your Load Balancer *elb-prod-guestbook*, select the Guestbook's VPC and create two listeners for your load balancer

would be as simple as modifying the guestbook application to redirect the traffic permanently (HTTP status code 301) to the HTTPS address.

After setting up the listeners, it is time to select the subnets. Since our application will be accessed by anyone, we will select the public subnets to deploy our Load Balancer to (select all subnets with *snpu* prefix) (see Figure 11).

Click on **Next: Assign Security Groups** to proceed. When asked to select a security group, select the security group with name *seg-us-east-1-elb:guestbook* and hit **Next: Configure Security Settings**. On this page, there are currently three ways to set up an SSL certificate on the Load Balancer:

- Choose an existing certificate from AWS Certificate Manager (ACM)
- Choose an existing certificate from AWS Identity and Access Management (IAM)
- Upload a new SSL certificate to AWS Identity and Access Management (IAM)

I chose to get a certificate from AWS Certificate Manager because it is free and you don't have to leave the AWS platform to deal with it, - which is very convenient. However, if you prefer to get a certificate from GoDaddy or any other Trusted Certificate Authority, you will see to manually upload the certificate to AWS.

If you have never requested a certificate on AWS before, check this quick tutorial: http://docs.aws.amazon.com/acm/latest/userguide/gs-acm-request.html

I will use a subdomain to point at the guestbook application, and since I own the domain **renandias.guru**, I requested a certificate for ***.renandias.guru**. If you select the ACM option, you can choose your certificate from the drop-down list (see Figure 12).

Next is one of the most crucial configurations: the ELB Security Policy. An ELB Security Policy contains SSL Protocols, Options and Ciphers to be used by the ELB. If you do not configure the policy correctly, you will have a lot of vulnerabilities in your infrastructure. So let's do it carefully!

Let's start with the SSL/TLS protocols. SSLv3 or earlier and TLS1.0 contain serious exploits such as POODLE

Available subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊕	us-east-1a	subnet-4fa8b606	172.20.48.0/20	snpr-us-east-1a-prod
⊕	us-east-1b	subnet-bf290de4	172.20.64.0/20	snpr-us-east-1b-prod
⊕	us-east-1c	subnet-e071e285	172.20.80.0/20	snpr-us-east-1c-prod

Selected subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊖	us-east-1a	subnet-80abb5c9	172.20.0.0/20	snpu-us-east-1a-prod
⊖	us-east-1b	subnet-f8290da3	172.20.16.0/20	snpu-us-east-1b-prod
⊖	us-east-1c	subnet-2271e247	172.20.32.0/20	snpu-us-east-1c-prod

Figure 11. The public subnets to deploy our Load Balancer

Step 3: Configure Security Settings

Select Certificate

An SSL Certificate allows you to configure the HTTPS/SSL listeners of your load balancer. You may select an existing SSL certificate or create a new one below. Learn more about setting up HTTPS load balancers and certificate management.

Certificate type:
- Choose an **existing** certificate from AWS Certificate Manager (ACM)
- Choose an **existing** certificate from AWS Identity and Access Management (IAM)
- Upload a **new** SSL certificate to AWS Identity and Access Management (IAM)

Request a new certificate from ACM
AWS Certificate Manager makes it easy to provision, manage, deploy, and renew SSL Certificates on the AWS platform. ACM manages certificate renewals for you. Learn more

Certificate: *.renandias.guru (1f0fef1a-7be8-45b5-860d-b074788c23cc)

Figure 12. The ACM option

and Heartbleed. For that reason, the PCI SSC (Payment Card Industry Security Standards Council) is requesting that systems using SSLv3 or earlier or TLS1.0 should use a secure version of TLS: TLS1.1 or higher. You can read more about this here: https://blog.pcisecuritystandards.org/migrating-from-ssl-and-early-tls

Back to the AWS console, select **Custom Security Policy**, and make sure **Protocol-TLSv1** is **not** selected. Although TLSv1.1 is more secure than TLSv1, TLSv1.1 is starting to be flagged as a vulnerability when running vulnerability scanners against your infrastructure. For this reason, unselect **Protocol-TLSv1.1** and leave only **Protocol-TLSv1.2** selected:

Then, make sure that the **Server Order Preference** option is selected. Here is AWS' explanation about this option:

"During the SSL connection negotiation process, the client and the load balancer present a list of ciphers and protocols that they each support, in order of preference. By default, the first cipher on the client's list that matches any one of the load balancer's ciphers is selected for the SSL connection. If the load balancer is configured to support Server Order Preference, then the load balancer selects the first cipher in its list that is in the client's list of ciphers. This ensures that the load balancer determines which cipher is used for SSL connection. If you do not enable Server Order Preference, the order of ciphers presented by the client is used to negotiate connections between the client and the load balancer."

Since we are not sure about which SSL Ciphers the client will offer, the best and safest way is to make sure the ELB will choose the Cipher.

Next, select the following SSL Ciphers:

ECDHE-ECDSA-AES128-GCM-SHA256

ECDHE-RSA-AES128-GCM-SHA256

ECDHE-ECDSA-AES128-SHA256

ECDHE-RSA-AES128-SHA256

ECDHE-ECDSA-AES128-SHA

ECDHE-RSA-AES128-SHA

ECDHE-ECDSA-AES256-GCM-SHA384

ECDHE-RSA-AES256-GCM-SHA384

ECDHE-ECDSA-AES256-SHA384

ECDHE-RSA-AES256-SHA384

ECDHE-RSA-AES256-SHA

ECDHE-ECDSA-AES256-SHA

AES128-GCM-SHA256

AES128-SHA256

AES128-SHA

AES256-GCM-SHA384

AES256-SHA256

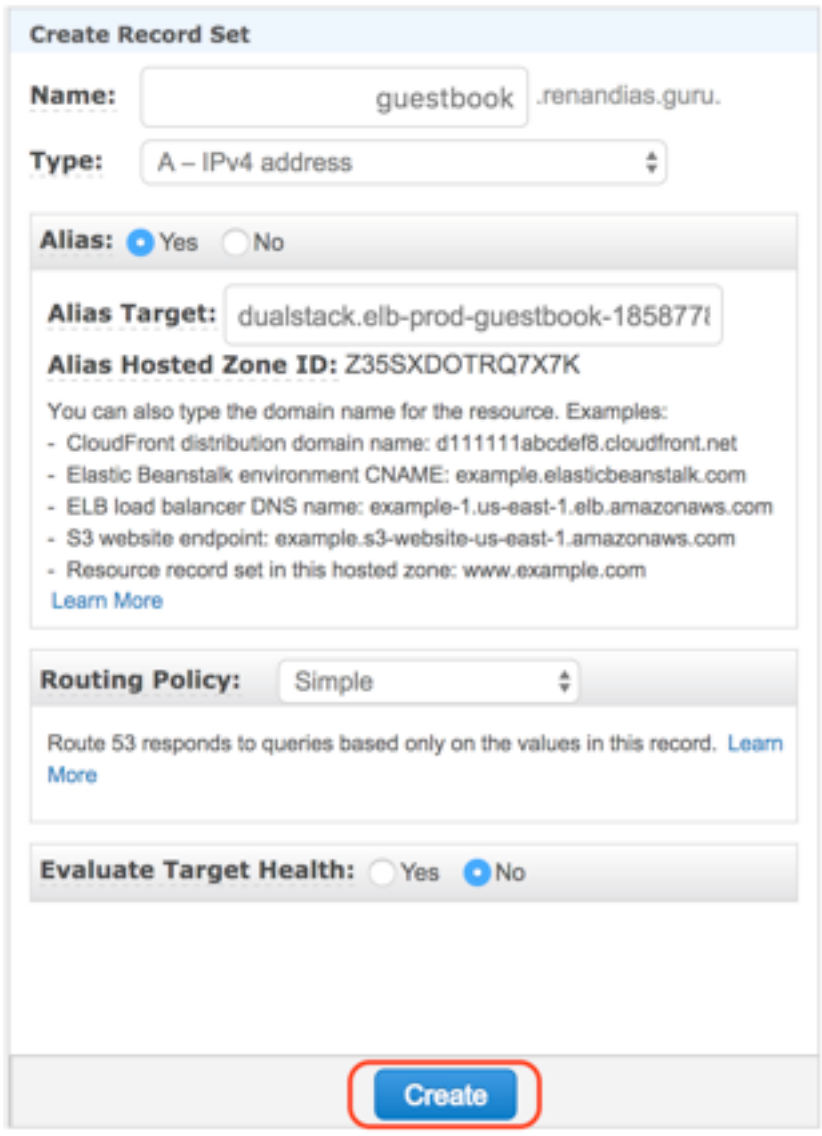
AES256-SHA

You will notice that we will not use the DHE-RSA-AES128-SHA cipher. The reason why is because this cipher contains a vulnerability called SWEET32 (Birthday attacks on 64-bit block ciphers). Read more about SWEET32 here: https://sweet32.info/

After you are done, click on **Next: Configure Health Check**. For the health check, we will use the **/health** endpoint:

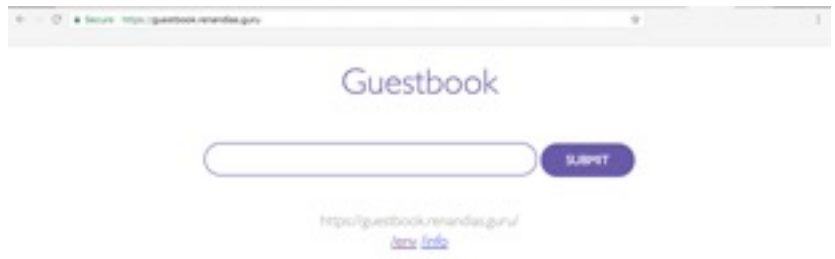
Click on **Next: Add EC2 Instances**. Now, find your guestbook instance and select it. Then, skip the step of adding tags and create the Elastic Load Balancer. After a minute, the Load Balancer will have a healthy instance and will start forwarding traffic to it (see Figure 13).

Hold on, we are nearly there! The very last step will be to create a DNS record to our application. Go to Route53 and click on the hosted zone in which you'd like to create the DNS record.



Create a DNS record such as **guestbook.yourdomain.com.** Then, select **A - IPv4 address** as Type. Now click on Alias **Yes** and paste the ELB's **DNS name**.

Type in on your browser the URL you have chosen for the guestbook application (remember to type in HTTPS before the URL):



Voilà! You have now created a quite complex AWS infrastructure which follows some of the PCI DSS security standards!

But… wait! There are a few more requirements that we have just complied with.

Compliance achieved: requirement 2.2.1a, 4.1e, A2.2

Is only one primary function implemented per server, to prevent functions that require different security levels from co-existing on the same server? (2.2.1a)

All the servers in our infrastructure implement only one primary function:

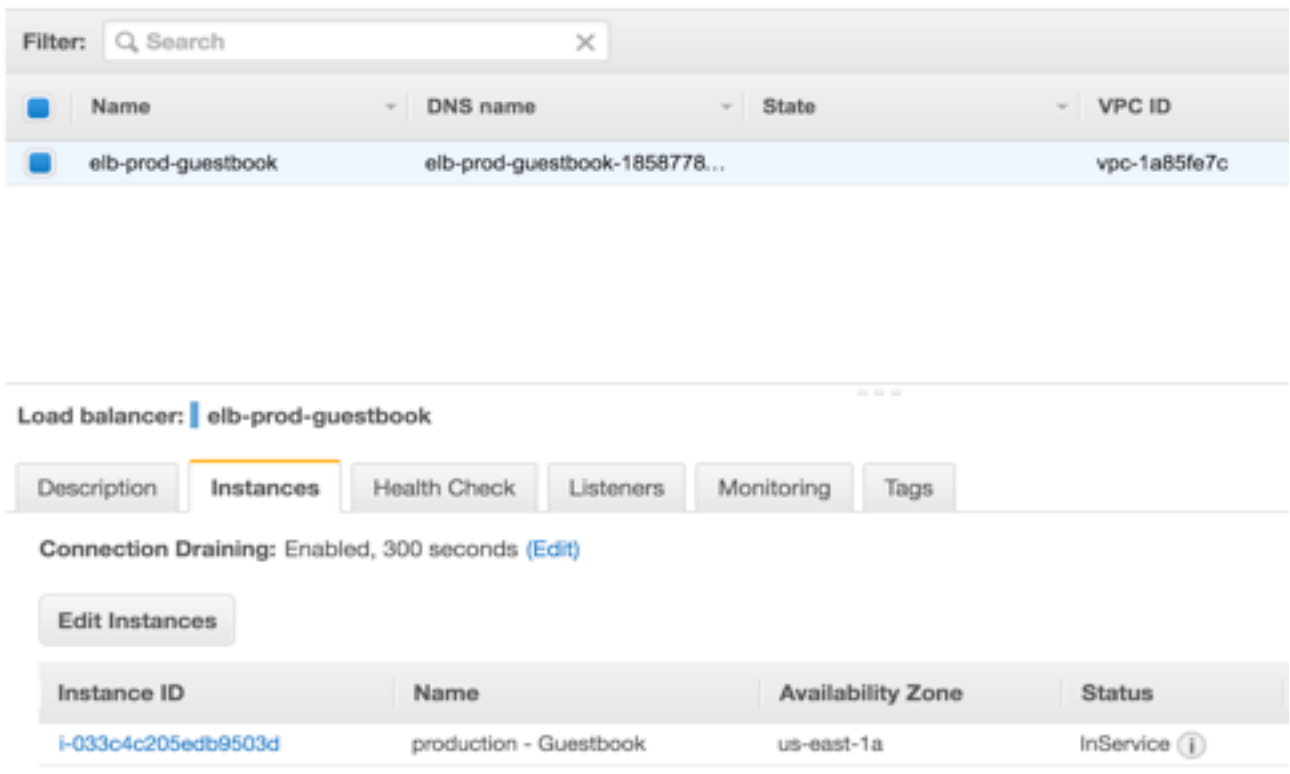


Figure 13. The Load Balancer will have a healthy instance and will start forwarding traffic to it

The Elastic Load Balancer only forwards traffic to the guestbook instance;

The Guestbook instance only serves the guestbook application;

The Bastion host is only used as a gateway for us to SSH into the guestbook instance;

The ElastiCache cluster is only running Redis;

And Amazon Route53's main purpose is around DNS

For TLS implementations, is TLS enabled whenever cardholder data is transmitted or received? (4.1e)

We are currently not transmitting or receiving any cardholder data, but because we have set up TLS on the Load Balancer, it would be really easy to comply with this requirement, right?

Is there a formal Risk Mitigation and Migration Plan in place for all implementations that use SSL and/or early TLS (other than as allowed in A2.1), that includes:(A2.2)

Our Elastic Load Balancer does not use SSL or early versions of TLS, which means that we do comply with this requirement. If your infrastructure were being assessed, you would indicate that this requirement is not applicable.

Other easily achievable requirements

Are the following audit trail entries recorded for all system components for each event:

User identification? (10.3.1)

Type of event? (10.3.2)

Date and time? (10.3.3)

Success or failure indication? (10.3.4)

Origination of event? (10.3.5)

Identity or name of affected data, system component, or resource? (10.3.6)

To keep track of what is happening on your system and comply with the requirements above, you can use the **Auditd** tool: https://linux.die.net/man/8/auditd

Are audit logs retained for at least one year?(10.7b)

To comply with requirement 10.7b, you would need to backup the log files generated by **Auditd**. You could, for instance, send them to S3 and store them for at least a year.

Are at least the last three months' logs immediately available for analysis? (10.7c)

Since PCI DSS requires that at least the last three months of logs are immediately available, you could send the log files to a standard S3 bucket. Then, you would create an S3 Lifecycle rule where these log files would be sent to either an infrequent access bucket or to Glacier - to save you some money.

Are configuration standards developed for all system components and are they consistent with industry-accepted system hardening standards? (2.2a)

Sources of industry-accepted system hardening standards may include, but are not limited to, SysAdmin Audit Network Security (SANS) Institute, National Institute of Standards Technology (NIST), International Organization for Standardization (ISO), and Center for Internet Security (CIS).

To comply with this requirement, you will need to configure your system according to at least one of these institutions. To give you an example, you can get Benchmarks documents from Center for Internet Security (CIS) on this link: https://learn.cisecurity.org/benchmarks

You only need to fill in the form and you will have access to tens of Benchmarks. You will find Benchmarks of **desktops and web browsers** (e.g. Apple OS X, Google Chrome, Internet Explorer, Firefox, Opera etc), **mobile devices** (e.g. iOS and Android), **network devices** (e.g. Cisco Firewall, Cisco IOS, Juniper JunOS), **operating systems** (e.g. Amazon Linux, CentOS, Ubuntu, Oracle Linux, Windows Server), **web servers and databases** (e.g. Apache Tomcat, Microsoft IIS, MySQL, MongoDB, Oracle Database server, SQL Server), **virtualization platforms and cloud** (Docker, VMware, Xen, Amazon Web Services) and **Microsoft office suite**.

Conclusion

As I mentioned before, PCI is bigger than just a couple of security standards. But I hope that this article gave you the feeling of how your infrastructure should be configured to be PCI compliant. If you wish to know more about PCI compliance on AWS, take a look at this very detailed guide: https://d0.awsstatic.com/whitepapers/compliance/AWS_Anitian_Workbook_PCI_Cloud_Compliance.pdf

Whether or not you are trying to get an Attestation of PCI Compliance from a Qualified Security Assessor (QSA), the tips on this articles will definitely get you started towards building a very secure infrastructure.

About the Author

Renan Dias is a passionate DevOps engineer that enjoys learning and writing about Amazon Web Services, infrastructure engineering and automation, HADR at scale, information security, continuous integration and deployment, and distributed systems. In the past, he worked with a wide range of technologies, such as AngularJS, Node.js, MongoDB, Meteor.js, Ionic, HTML/CSS, LAMP stack, iOS SDK, OpenCV, Java and MATLAB. He has also contributed to a Brazilian university's research about Computer Vision and received numerous awards during his student life. When not doing any infrastructure-related work, Renan spends his time traveling with his loved ones, learning languages (English, Spanish, German and Polish), practicing sports (Tennis, Ping Pong, Basketball, Cycling, Kung Fu) and playing the electric guitar and the drums.
You can find Renan on LinkedIn: https://www.linkedin.com/in/renan-dias-a2801163

AWS Infrastructure Security: Deep Dive into Access Control Management

You will learn ...

- How to think through the security of your AWS infrastructure.
- Some tools that AWS provides to improve the planning of Access Control.

You should be familiar with ...

- Security-related topics.
- AWS services.
- Access Control Management.

Introduction

A recent report indicates that the cloud market was valued at $148 billion in 2016 with expected annual growth rate at 25%. Statistics, provided by RightScale, show that 31% of enterprises are running more than 1000 VMs in private cloud and 17% of enterprises are having more than 1000 VMs in public cloud. Given such massively sophisticated infrastructures, it's crucial to design your cloud infrastructure efficiently. In the meantime, Amazon continues to dominate the cloud market with almost one-third of the market share. So, we will be focusing on the infrastructure of Amazon Web Services (AWS).

There are some tools that can be exploited to help designing efficient scalable AWS infrastructure. This article aims to discuss various considerations with access control management in AWS infrastructure. The weakest link in an enterprise is the people who administer and exploit the resources. Thus, it's important to learn the best practices in securing such activities/information from hacker hands and allow for the scalability of organization infrastructure.

Main Body

Let's start with discussing authentication. Authentication is a key corner in access control management. It is required by most credible security and compliance standards such as NIST, ISO and HIPPA. In AWS, the access is granted based on least privileges and this access is audited via SOC-2 audit on 24/7 basis. While planning for AWS authentication, it's critical to consider user-based and group-based security policies. AWS opens the doors for managed, inline or custom policies written in JSON format (Details of security policies are beyond the scope of this article). For such users and groups, it's always recommended to have password-policy that defines the criteria of strong passwords that should be used. Also, make sure to think through authorization levels from user, group and

resource perspectives while developing the security policies.

AWS supports six different authentication options. These options can be bundled or used individually:

Email Address and Password: this is the root-user account for the person who first creates the AWS account for the organization.

Multi-Factor Authentication (MFA): AWS supports MFA in which you will get a code on your registered phone number, for example, to confirm your identity.

Access Keys: can be used to provide third-party access or implement oauth-2 like mechanisms. This is important for automating processes. Please note that AWS limits the number of access keys to two.

Key Pairs: AWS supports public-key cryptography which allow for multiple users to have similar credentials. You can have maximum of two key-pairs. AWS supports CloudFront Key pairs and X.509 certificates. (X.509 certificates are digital certificates that use the X.509 public key infrastructure standard to associate a public key with an identity contained in a certificate).

IAM Usernames: Root user creates different user accounts for the users in the organization who are supposed to use different AWS services. These users will have controlled-customized access to resources.

Figure 1.0 shows a screenshot of “Security Credentials” page from Console login to AWS. Blue arrows refer to the different authentication options that can be configured. Although AWS provides console login for users, you should be wise in making the proper decision in selecting the authentication mechanism. For example, key pairs might be suitable to provide access to multiple users with similar credentials.

However, if access keys were poorly designed, you might end up with unauthorized users consuming the organization’s AWS services once they get access to these keys. In AWS, user and authentication related items are managed through Identity and Access Management (IAM). If you are not familiar with IAM, think about active directory in an organization. IAM is an improved version of active directory to manage users, groups, roles, policies and various account settings in AWS.

That is all about user accounts but what about the root-level account?! Amazon refers to the first AWS account that gets created for the organization on AWS as “Root Account”. This account has full access to all account resources; it can’t be disabled and it can’t be controlled via IAM. Please remember that this account should NOT be used for daily operation. Login information for the root account should be stored in an encrypted format in a “safe” place. This place should have MFA mechanism applied to it. AWS supports Google Authenticator for virtual devices and Gemalto for hardware devices.

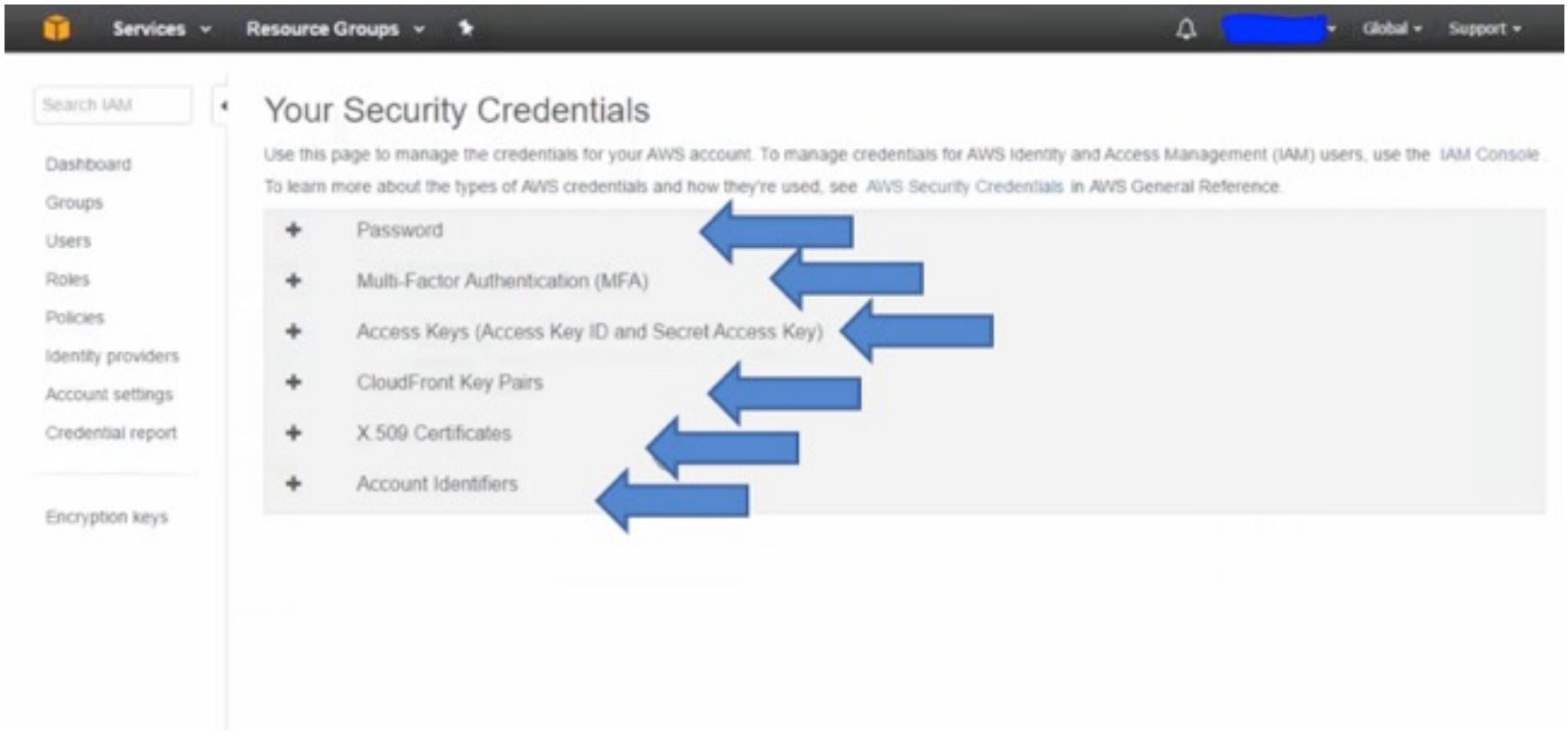


Figure 1.0 Authentication Options in AWS

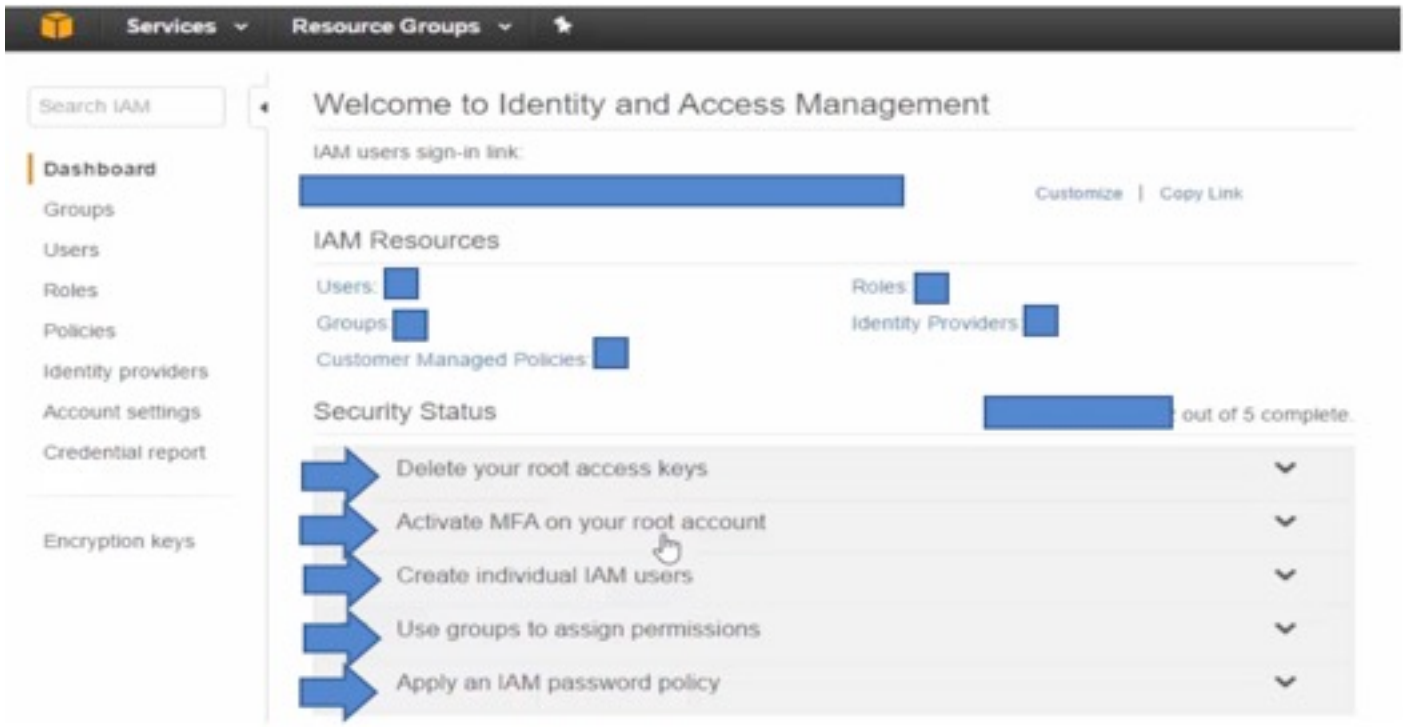


Figure 2.0 IAM Dashboard Page for Root Level Account

Due to the cost of maintaining stored login-information for the root-level account, some experts recommend that you don't have to store this login information!!! SMEs suggest that you can create an IAM role with full admin privileges and change the root account password to something you won't remember (e.g. random string). Once you change the password, don't save the new password and use a forget-password link to gain access to the root account if needed.

Typically, you should NOT use the root account unless you need to change the root account info, change billing info, buying new AWS items, testing AWS' public IP address, or close AWS account. Moreover, there are various security-related activities that can be done on the root account. These activities are monitored via the account dashboard. Ideally, the security status on the dashboard should show 5/5 completed activities. Figure 2.0 shows the dashboard of IAM for a Root-level account. Please note that you may not need MFA mechanism if you are going to change the account password to a random string.

Once the root account is properly configured, you should start thinking about creating user accounts. Generally, it's recommended to have an account for each environment for your applications (e.g. development, test and production). Furthermore, you may need additional accounts for more granular security or specific needs (No worries, you can enable cross account access to use resources shared by other accounts). Keep in mind that the more accounts you have, the better granularity of

Figure 3.0 IAM Create Role Page

user account security but also the higher complexity user management becomes. If you encounter unexpected high bills, you can enable consolidated billings on your accounts to monitor the billing activity of each account separately.

One way to think about designing multiple user accounts is to consider these accounts from governance perspective in which you have trusting account (provider) and a trusted account (consumer). Trusting accounts should have higher privileges to support the creation of Virtual Private Clouds (VPCs), Network Access Control Lists (NACLs), Subnets, consumer accounts, IAM, managed services, Virtual Private Networks, security groups and EC2 instances.

On the other hand, the trusted account will only focus on the resources that can be consumed (e.g. EC2 Resources, database resources (RDS) or Elastic Boxes).

AWS makes it easy for you to apply the provider/consumer concept by providing the ability to create Roles, Groups and Users through IAM. During the creation of the new role, you will have the option to define the security policy(ies) that can control this role. Similarly, groups and users can be configured. Figure 3.0 shows the five steps that are needed to create a new customized role.

There are various approaches to access resources supported by AWS. These approaches can be categorized into administrative (AWS management console and Command Line Interface), APIs and Managed Services. Details on these items are beyond the scope of this article but it's important to note that these tools do not monitor all resource access control metadata.

Information related to Relational Database Server logon or EC2 instance logon is not captured. Such information can be monitored via OS-based monitoring or logging.

While creating roles, users and groups, think about your password management strategy. Password management is a vital element of secure infrastructure. You will need to consider aspects like minimum password length, password reuse, complexity rules, password expiration and expiration procedure.

AWS supports the use of various tools to collect statistics on password metadata. These tools include IAM console, AWS CloudTrail, Credential Report, Access Advisor, Simple Notification Service (SNS) and

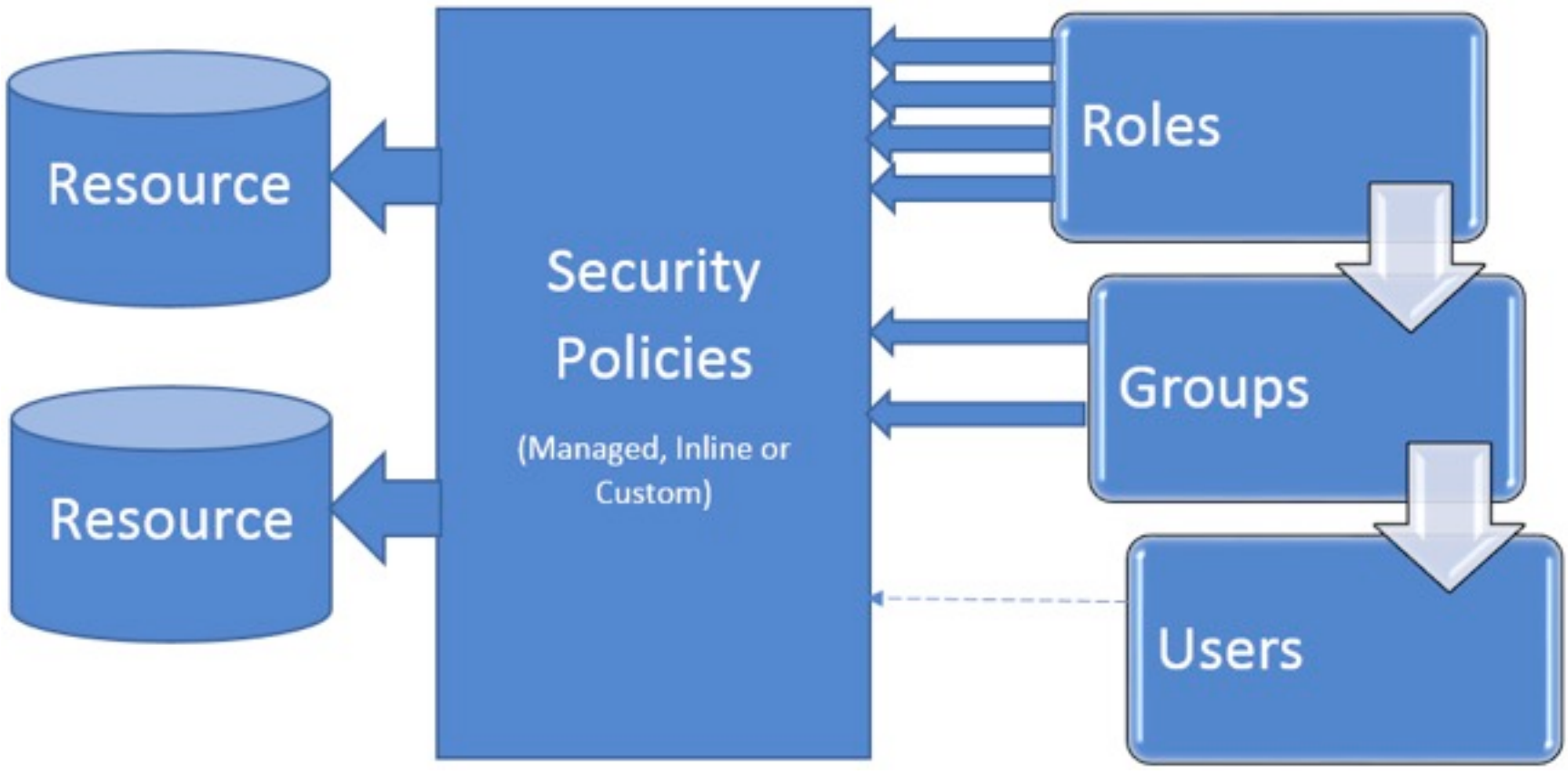


Figure 4.0 Best Practices in Designing AWS IAM Accounts for Organization

CloudWatch (Details on these tools are beyond the scope of this article). The rule of thumb is that if you are at the console, you need a password. If you are consuming an API or using CLI, you need access keys.

Figure 4.0 summarizes best practices needed to be considered for designing scalable IAM accounts. Please notice the density of rows originated from Roles and Groups. It's expected that policies are applied at a more generic level so they can be easily maintained.

Users are not expected to have special permissions except in extreme scenarios where one user has ultimate special needs that are not shared by other users.

Conclusion

Finally, we can summarize the best practices that we've discussed in this article in the following bullet points:

Individual Users:

- Always start by granting least-privileges.
- Enable Credential Rotation per user.
- Use unique Credentials for each User.

Permissions:

- Manage Permissions through Roles and Groups.

Conditional Access:

- Restrict Database creation to be limited to specific engine.
- Restrict access from specific IP address/range.

Auditing:

- Enable CloudTrail to log API calls activities.
- Log calls to S3 bucket associated with your account.

Strong Password:

- Use password tools.
- Consider Password Strength Metrics.

Credential Rotations:

- Rotate your passwords on fixed schedule.
- Use Credential Report to identify credentials should be rotated.
- Note that EC2 instances rotate credentials for IAM roles by default.

MFA:

- Enable MFA for all users.

Sharing:

- Use IAM roles to share access. (Use IAM roles for EC2 instances).

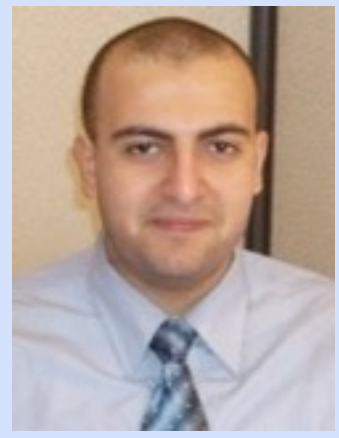
Sources of Provided Statistics

http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2016-state-cloud-survey

http://www.waterfordtechnologies.com/cloud-computing-stats-2017/

https://www.forbes.com/sites/benkepes/2015/05/22/how-are-organizations-using-amazons-cloud/#4ee7011c3e29

About the Author



Mohamed Farag has been in the IT industry for 7 years out of which 4 years were spent in business consulting. Mohamed is Salesforce Certified Platform Developer, Certified SAFe 4.0 Agilist, Certified IBM Mobile Application Developer and Certified Microsoft Technology Specialist. On part-time basis, Mohamed pursues his PhD in the field of Systems Engineering at George Washington University.

Conventions over Restrictions – Programming the Python Way

BY MIKE MÜLLER

Python is a powerful programming language. It supports procedural and object-oriented programming, and provides important features for functional programming. Its dynamic typing and many meta-programming features allow doing nearly everything at run time. While this freedom provides lots of opportunities for elegant solutions, Python might also be perceived as a dangerously unrestricted language. The common solution is to use conventions to solve a problem in the preferred, hence, “pythonic” way. Experienced Python programmers tend to stick to these conventions as much as possible and only diverge when the benefits are substantial compared to that of the conventional solution.

This article gives a short overview of Python features focusing on conventions and their benefits. Since Python syntax is often called executable pseudo code, a reader with solid programming experience does not need to have previous knowledge of Python to follow along.

Password Cracking in UNIX

Passwords are used for performing authentication. The system can be authenticated using different ways like something which a user knows (passwords), something that the user has (identification token), or something which the user is (biometric).

Out of the three things listed above, the password can be changed easily in case one finds that the same has been compromised. In this paper, we will talk about various password cracking tools available for cracking passwords in a UNIX environment. Password cracking can occur if the cracker gains access to the system through physical or remote access. The cracker could attempt to try each possible password combination. If the attacker gains access to hashes of the passwords, then it is possible to use software which utilizes Rainbow tables to crack passwords.

Password Storage

The file system in UNIX environment is secure with permissions at user/group/everyone level. Further, the password is stored in hash format. Whenever the user enters the password, the system converts the password in hash format and checks against the hash already stored in the password file.

The password file in Unix-based system is located at /etc/passwd. This file is readable by any user but can be written only by superuser. This file contains one line per account in the system. Each entry in a password file has seven (7) fields. The first field specifies the account / username. The second field has the letter ‘x’ written. This was used earlier to store the hash. However, in a newer version of Unix, the hash is stored in a shadow file located at /etc/shadow which is readable by those having superuser privileges.

Password Strength

According to Wikipedia, password strength is a measure of the effectiveness of a password in resisting guessing and brute-force attacks.

In its usual form, it estimates how many trials an attacker would need, on average, to guess it correctly. The strength of a password is a function of length, complexity and unpredictability.

Using strong passwords lowers the overall risk of a security breach, but strong passwords do not replace the need for other effective security controls. The effectiveness of a password of a given strength is strongly determined by the design and implementation of the factors (knowledge, ownership and inherence).

The rate at which an attacker can submit guessed passwords to the system is a key factor in determining system security. Some systems impose a time-out of several seconds after a small number (e.g. three) of failed password entry attempts. In the absence of other vulnerabilities, such systems can be effectively secured with relatively simple passwords.

Hashing

Unix uses one-way hashing technique to store password.

```
[root@ac ~]# cat /etc/shadow
root:$1$Etg2ExUZ$F9NTP7omafhKIlqaBMqng1:15
651:0:99999:7:::
```

The password stored in /etc/shadow file has 3 fields:

1st Field: This is a numerical number which tells about the hashing algorithm being used.

- $1 = MD5 hashing algorithm.
- $2 = Blowfish Algorithm is in use.
- $2a = eksblowfish Algorithm
- $5 = SHA-256 Algorithm
- $6 = SHA-512 Algorithm

2nd Field: This is the salt value.

3rd Field: The last field is the hash value of salt+user password.

Attacks to Recover Password

Brute Force Attack: As per Wikipedia, a brute-force attack consists of an attacker trying many passwords or passphrases with the hope of eventually guessing correctly. The attacker systematically checks all possible passwords and passphrases until the correct one is found. Alternatively, the attacker can attempt to guess the key which is typically created from the password using a key derivation function. This is known as an exhaustive key search. A brute-force attack is a cryptanalytic attack that can, in theory, be used in an attempt to decrypt any encrypted data (except for data encrypted in an information-theoretically secure manner). Such an attack might be used when it is not possible to take advantage of other weaknesses in an encryption system (if any exist) that would make the task easier.

When password guessing, this method is very fast when used to check all short passwords, but for longer passwords, other methods such as the dictionary attack are used because a brute-force search takes too long. Longer passwords, passphrases, and keys have more possible values, making them exponentially more difficult to crack than shorter ones. Brute-force attacks can be made less effective by obfuscating the data to be encoded, making it more difficult for an attacker to recognize when the code has been cracked or by making the attacker do more work to test each guess. One of the measures of the strength of an encryption system is how long it would theoretically take an attacker to mount a successful brute-force attack against it. Brute-force attacks are an application of brute-force search, the general problem-solving technique of enumerating all candidates and checking each one.

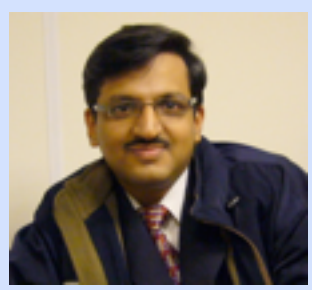
Dictionary Attack: As stated in Wikipedia, a dictionary attack is a technique for defeating a cipher or authentication mechanism by trying to determine its decryption key or passphrase by trying hundreds or sometimes millions of likely possibilities, such as words in a dictionary.

Rainbow Tables: A rainbow table is a precomputed table for reversing cryptographic hash functions, usually for cracking password hashes. Tables are usually used in recovering a plaintext password up to a certain length, consisting of a limited set of characters. It is a practical example of space/time trade-off, using less computer processing time and more storage than a brute-force attack which calculates a hash on every attempt, but more processing time and less storage than a simple lookup table with one entry per hash. Use of a key derivation function that employs salt makes this attack infeasible. Rainbow tables are an application of an earlier, simpler algorithm by Martin Hellman.

Password Cracking tools for Unix

- John the Ripper
    - John the Ripper is a fast password cracker, currently available for many flavors of Unix, Windows, DOS, and OpenVMS. Its primary purpose is to detect weak Unix passwords. Besides several crypt(3) password hash types most commonly found on various Unix systems, supported out of the box are Windows *LM hashes*, plus lots of other hashes and ciphers in the community-enhanced version.
    - It is free and Open-Source software, distributed primarily in a source code form.
- THC Hydra
    - THC Hydra is a proof of concept code, to give researchers and security consultants the possibility to show how easy it would be to gain unauthorized access from remote to a system.
- RainbowCrack
    - RainbowCrack is a general propose implementation of Philippe Oechslin's faster time-memory trade-off technique. It cracks hashes with rainbow tables.
    - It uses time-memory tradeoff algorithm to crack hashes. It differs from brute force hash crackers.

About the Author



Amit Chugh is creative, innovative & results driven technology leader with over 18 years of industry experience. Out of which Amit has spent 11+ years in Information Security Management, Incident Management, Business Continuity Management and Software Development. Amit is Certified Ethical Hacker, Certified ISO 27001 Lead Auditor, and Microsoft Certified Product Specialist.He is reachable at chugh ‘dot’ a ‘at’ gmail ‘dot’ com.

Elastix on Bhyve

What is Elastix?

Elastix is a unified communications server software that brings together IP PBX, email, IM, faxing and collaboration functionality. It has a Web interface and includes capabilities such as a call center software with predictive dialling.

The Elastix 2.5 functionality is based on open-source projects which includes Asterisk, FreePBX, HylaFAX, Openfire and Postfix. Those packages offer the PBX, fax, instant messaging and email functions, respectively.

As for Elastix 5.0, its functionality is provided through 3CX, a software based private branch exchange (PBX) built on the SIP (Session Initiation Protocol) standard. It enables extensions to make calls via the public switched telephone network (PSTN) or via Voice over Internet Protocol (VoIP) services. Elastix 5.0 is an IP business phone system that supports standard SIP soft/hard phones, VoIP services and traditional PSTN phone lines.

Elastix 2.5 is free software released under the GNU General Public License whereas Elastix 5.0 is Proprietary released under the terms of the 3CX license.

What is Bhyve?

Bhyve (pronounced "bee hive", formerly written as BHyVe) is a type-2 hypervisor/virtual machine manager for FreeBSD that was introduced in FreeBSD 10.0, and supports most Intel and AMD processors that report the "POPCNT" (POPulation Count) processor feature in dmesg(8).

The Bhyve BSD-licensed hypervisor became part of the base system with FreeBSD 10.0-RELEASE. This hypervisor supports a number of guests, including FreeBSD, OpenBSD, and many Linux distributions.Virtualization offload features of newer CPUs are used to avoid the legacy methods of translating instructions and manually managing memory mappings.

The Bhyve design requires a processor that supports Intel Extended Page Tables (EPT) or AMD Rapid Virtualization Indexing (RVI) or Nested Page Tables (NPT).

Currently, Bhyve can run the following guests: FreeBSD 9+, OpenBSD, NetBSD, Linux and MS Windows desktop (versions Vista, 7, 8/8.1/8.2 and 10), as well as MS Windows Server (versions 2008/2008R2, 2012/2012R2 and 2016 Technical Preview 2 and 3).

Lately, libvirt supports Bhyve as well. libvirt is an open source API, daemon and management tool for managing platform virtualization. This tool is used to manage KVM, Xen, VMware ESX, QEMU and other virtualization technologies. You can use virt-manager to manage Bhyve, but personally I prefer to utilize Bhyve from shell. Moreover, you can choose other FreeBSD packages which were created to make life easier — like CBSD and VM-Bhyve.

Recently, Bhyve supports Unified Extensible Firmware Interface Graphics Output Protocol or "UEFI-GOP" which means that you can run any modern OS without pain.

Bhyve Preparation and Elastix Installation

Elastix requirements:

- Minimum required RAM is 2 GB.
- Minimum recommended virtual disk size of 30GB.

Install FreeBSD 11.0: You can also install FreeBSD 11.0 or any latest builds.

Install Grub-emu loader for Bhyve: We must install the “grub2-bhyve” port. This process is very time-consuming and needs user-interaction. But with some tricks, we can do it very easily:

```
# cd  /usr/ports/sysutils/grub2-bhyve

# make install clean -DBATCH
```

-DBATCH force port building process to not prompt you for confirmation and do it automatically.

Hypervisor, Network and Storage Preparation:

 # kldload vmm — this command will load Bhyve kernel module or driver.

 # ifconfig tap0 create up — this command creates a new interface and brings it up.

 # ifconfig bridge0 create up — this command also creates a bridge and makes it up and ready.

 # ifconfig bridge0 addm em0 — this command adds em0(network interface) to bridge0

 # ifconfig bridge0 addm tap0 — this command adds tap0 to bridge0.

 # truncate -s 30G elastix.img — this command creates a file with 30GB size.

Prepare Elastix ISO:

```
#fetch
https://excellmedia.dl.sourceforge.net/project/elastix/Elastix%20PBX%20Appliance%20Software/2.5.0/latest/Elastix-2.5.0-STABLE-x86_64-bin-08may2015.iso

# mv
Elastix-2.5.0-STABLE-x86_64-bin-08may2015.iso elastix.iso
```

Create a elastix.map that grub will use to map the virtual devices to the files on the host system:

```
# touch elastix.map

# echo "(hd0) /root/elastix.img"  >>
elastix.map

# echo "(cd0) /root/elastix.iso"  >>
elastix.map
```

Boot Elastix Virtual Machine:

```
# grub-bhyve -m elastix.map -r cd0 -M
2048 elastix

          grub> linux
(cd0)/isolinux/vmlinuz

          grub> initrd
(cd0)/isolinux/initrd.img

          grub> boot
```

```
#bhyve -A -H -P -s 0:0,hostbridge -s
1:0,lpc -s 2:0,virtio-net,tap0 -s
3:0,virtio-blk,elastix.img -s
4:0,ahci-cd,elastix.iso -l com1,stdio -c 2
-m 2048M  elastix
```

this command makes a virtual machine(elastix) with 2 cores CPU and 2G of ram.

-H Yield the virtual CPU thread when a HLT instruction is detected. If this option is not specified,virtual CPUs will use 100% of a host CPU.

-A Generate ACPI tables that required foramd64 guests.

-P Force the guest virtual CPU to exit when a PAUSE instruction is detected.

other parameter’s define CDROM and HDD.

Elastix installation: You can install Elastix with the GUI wizard.

Elastix First Boot

After the installation of Elastix, the system will request a reboot. This reboot causes Bhyve to exit.

Issue these commands to boot Elastix again:

```
#bhyvectl --destroy –vm=elastix

#grub-bhyve -m elastix.map -r hd0,msdos1
-M 2048M elastix

     linux
     (hd0,msdos1)/vmlinuz-2.6.18-371.1.2.e
     l5
     root=/dev/mapper/VolGroup00-LogVol00

     initrd
     (hd0,msdos1)/initrd-2.6.18-371.1.2.el
     5.img

     boot

#bhyve -A -H -P -s 0:0,hostbridge -s
1:0,lpc -s 2:0,virtio-net,tap0 -s
3:0,virtio-blk,elastix.img -l com1,stdio
-c 2 -m 2048M elastix
```

As you can see, Elastix will boot and welcome will show us the IP address of Elastix Web GUI. However, this address doesn't work (I gave elastix 192.168.1.20). Why?

IPTables (the linux firewall) is running and you must stop it to communicate with Apache. So, issue the following command:

```
     #service iptables stop
```

You can also create a IPTAbles rule to bypass any port but it's better to use host firewalling and disable any guest firewall in virtual infrastructure.

Now, you can see the Elastix Web GUI but something's still wrong. Elastix doesn't allow changing of the configuration. This is because of SELinux.

Secret Sauce

Security-Enhanced Linux (SELinux) is a Linux kernel security module that provides a mechanism for supporting access control security policies, including United States Department of Defense–style mandatory access controls (MAC).

SELinux is a set of kernel modifications and user-space tools that have been added to various Linux distributions. Its architecture strives to separate enforcement of security decisions from the security policy itself and streamlines the volume of software charged with security policy enforcement. We have two solutions:

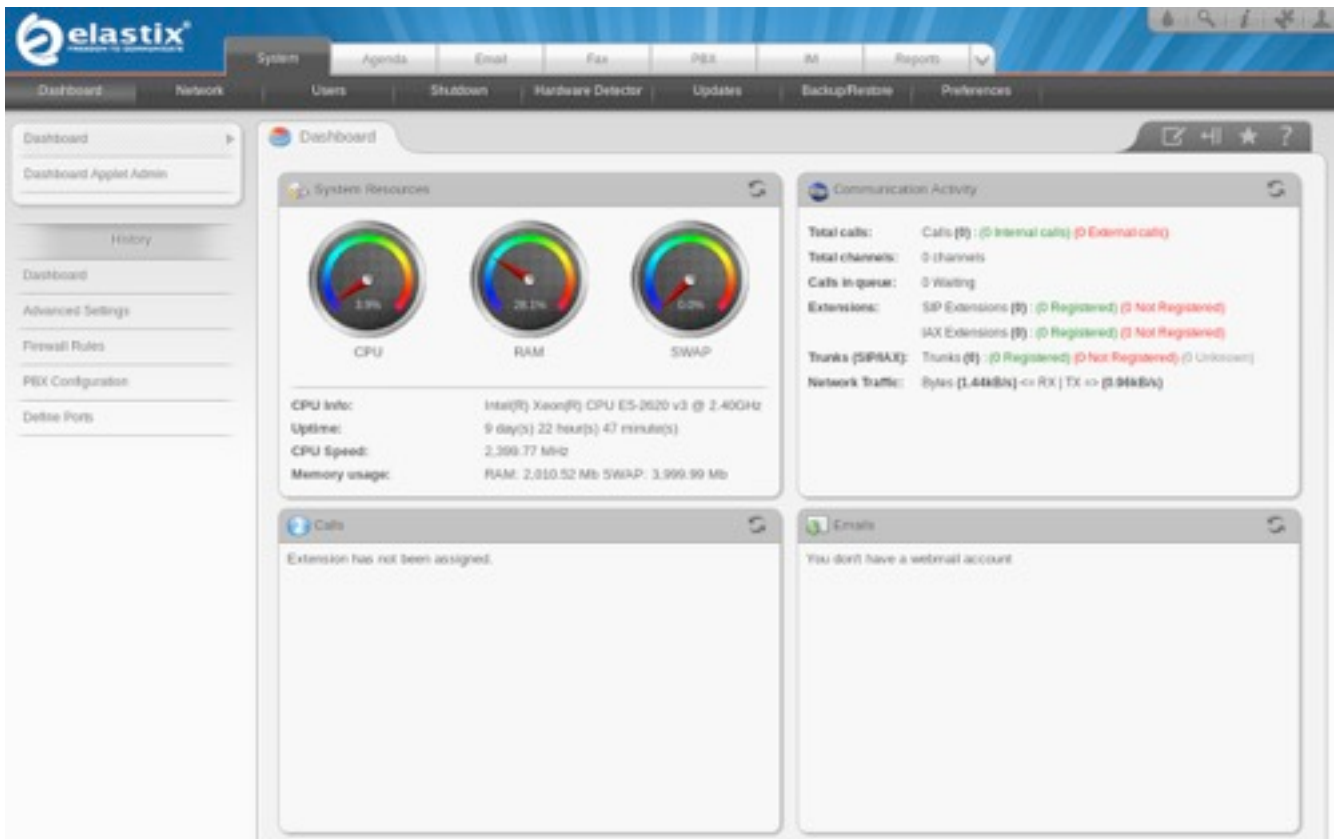


Figure 1. Elastix

Completely turning off SELinux at /etc/selinux/config. You need to change the SELINUX option to disabled: *SELINUX=disabled*

Configuring SELinux to log warnings instead of block at /etc/selinux/config. You need to change the SELINUX option to permissive: *SELINUX=permissive* and then issue this:

```
#setenforce 0
```

And it's done, Elastix is now up and running (see Figure 1.)

Make Config Persistence

Create a file and name it vml:

```
#touch vml
```

Open vml with **ee** and paste these commands to vml:

```
#!/bin/sh

. /etc/rc.subr

name=vml

rcvar=vml_enable

start_cmd="${name}_start"

stop_cmd=":"

load_rc_config $name

: ${vml_enable:=no}

: ${vml_msg="Nothing started."}

vml_start()

{

     kldload vmm

     ifconfig tap0 create up

     ifconfig bridge0 create up

     ifconfig bridge0 addm em0

     ifconfig bridge0 addm tap0

}

     run_rc_command "$1"
```

Copy vml to /etc/rc.d:

```
#cp vml /etc/rc.d/
```

Make it executable:

```
#chmod +x /etc/rc.d/vml
```

Add vml script to /etc/rc.conf:

```
#echo 'vml_enable="YES"'  >> /etc/rc.conf
```

So after rebooting the host machine, vml script will initiate Bhyve configuration.

Conclusion

Elastix installation is easy. Though, if you want to use FXO/FXS PCI-E hardware, you have to know about Bhyve PCI Passthrough. The Bhyve hypervisor supports the passing of PCI devices belonging to the host through to a virtual machine for its use, exclusively. In future, we will talk about Bhyve PCI Passthrough and explain how to attach one device to specific Bhyve VM.

Useful Links

https://wiki.freebsd.org/bhyve/pci_passthru

https://www.amazon.com/Analog-Express-Connector-Elastix-Freepbx/dp/B00IK7F7KI

https://www.elastix.org/docs/

https://wiki.freebsd.org/bhyve

http://in4bsd.com/page/FreeBSD

About the Author

Abdorrahman Homaei has been working as a software developer since 2000. He has used FreeBSD for more than ten years. He became involved with the meetBSD dot ir and performed serious training on FreeBSD. He is starting his own company in Feb 2017.

You can visit his site to view his CV: http://in4bsd.com

Ready to Land on IoT World with the Raspberry Pi 3

In the recent years, the small boards with that red fruit logo have become very popular. Before the “Pi”, those boards were known almost exclusively in university environments where they were used to learn to program, create small research projects, etc. With the emergence of Raspberry and their “Pi” board, the cost and ease of access to these devices has been reduced in such a way that uses and applications have exploded and seem to have no ceiling. In addition, the company has made a very appropriate approach, facilitating the adoption since the beginning (with tutorials, helps, images, forums etc.) and showing a clear roadmap where we see possibilities for our projects are nearly limitless with the Pi.

There are also some utilities like NOOBS (New Out Of Box Software) to help newbies with questions like, *“I already have it, and now what?”*

I acquired the Raspberry Pi 3 (on the picture) a few months ago. It’s not their latest model, where again, the power and speed have been slightly increased. In my Pi 3 model, the main improvement in face of the previous versions was the inclusion of WiFi and Bluetooth modules on the same board, which made no longer necessary to connect a USB to provide these capabilities. That meant easiness but also a reduction in need of power supply. It’s not just the manufacturer and the fan community bringing new developments, also big companies like Microsoft, IBM or Ubuntu are making contributions and encouraging the use of these small devices.

What is the main objective? First of all and thanks to the easiness to add different sensors, the Internet of Things (or *IoT*), could be one of them, then automation, gaming (specially retro),… and for sure always learning.

From the research I’ve done these past months, we’ve seen new versions of:

Raspbian -> it’s a Linux distribution based on Debian. It’s the most popular within the Raspberry Pi world. The last version was called Raspbian Jessie and includes Pixel, a new graphic interface optimized for the Raspberry Pi. Pixel stands for “**P**i **I**mproved **X**windows **E**nvironment, **L**ightweight”.

IBM -> Watson IoT & Bluemix for Raspberry Pi Great for connecting services oriented to IoT.

Microsoft -> Windows IoT Core for Raspberry Pi A new version of the popular OS thought for IoT.

Ubuntu -> Ubuntu Mate for Raspberry Pi The version of this known Linux distro adapted to the Pi.

Since **I have no previous experience with programming**, I have mainly focused on the process of searching for information, reading manuals, watching video tutorials and examples on Youtube etc. But certainly, one of the great attractions that I see is to do with my automation project, and have it connected to the cloud or Watson and provide it with some artificial intelligence (AI).

As you can see, I’m still in my infancy around Raspberry Pi, but the ideas don’t stop coming, so I intend to share my advances. I have also recorded and uploaded to my YouTube channel the first boot of the Raspberry Pi 3 in Raspbian with a graphical interface. The Raspberry Pi is OS agnostic, meaning that you can use it with the OS that you prefer (different Linux distros, Windows etc.). In my learning and tests, I’m using Raspbian as I mentioned. Most of the steps and commands will be very similar in other Linux distros. I’m learning and doing these experiences during my spare time. Thus, my progress is slower than I expected. Anyway, let’s put some meat to this article. Here are the first steps, “from newbie to newbie” ;-)

Be sure to have a keyboard and mouse so that you can use them for the Raspi. At least for the initial configuration, since you'll see that afterwards, it's quite easy/comfortable to access it remotely.

Take a microSD card (the size will depend on what you want to do, a minimum of 4GB), the faster the better. Install the Operating System (I'm using Raspbian) on the card. There are great official guides for that. Once finished, insert the card in the Raspi.

Plug the Raspi to a screen through HDMI, to the internet through an ETH cable and to a power source using the microUSB port. You can use the keyboard and mouse either through USB or Bluetooth. Allow it to boot the system and complete the initial setup.

In case you had any issue with screen, run "*raspi-config"* from the terminal. It's the main configuration tool in Raspbian. Check options for keyboard, screen, etc. On the "Advanced options", be sure you enable "SSH". This is the way we will use later on to access remotely.

If you don't have a ETH cable at a hand, once booted, I suggest to configure the Wi-Fi. If you're a newbie as I am, you probably have chosen to start with graphical interface. So, configuring Wi-Fi is very simple. The same way you would do on your desktop PC, clicking on the "Wireless" logo.

Test the internet connection. Open a Terminal and perform a *system upgrade*. It's always better to have the last version of the system and packages we will be using. The *root* password is initially disabled by default. Use *"sudo su"* to gain *root* access into the terminal or simply add "*sudo*" before the commands.

```
sudo apt-get update
```

This will update the lists from which our system is taking the different packages.

```
sudo apt-get dist-upgrade
```

This instruction will update all the installed packages (including the system kernel) to their latest version. Once the system is updated and running, we can finally start using it. Up to now, I have completed three different mini-projects that I will describe later. Since I have different microSD cards available, I saved those "projects" on a different card, each one. Quick presentation:

Learning to code and tests: It's my main card. I have the latest Raspbian version with the different tools I'm using to learn how to do things with the Raspi, and to code. For the moment, I'm trying to learn Python, although I have also tried a little bit of Node-Red. Accessing through SSH is key, since I just have to supply power to my Raspi and gain access from any PC, mobile or tablet via Wi-Fi **in a secured way**. You always have the possibility of accessing via a Remote Desktop, with graphic interface. However, it's much slower and insecure (since the encryption is not included natively).

Gaming center: Do you remember those old days with the NES? The NES (if you don't know it) was a classic Nintendo game console. I do remember it, specially on Christmas. This year, I had again a retro console in Christmas ☺ For that, I used a customized version of Raspbian called "RetroPie" and connected my PS3 controls through Bluetooth.

Media center: I already have an Android box with Kodi and other apps that work very well. But since one of the most common uses for the Raspi is a home media center, I wanted to try it out. I chose another Raspbian modified version called LibreElec, whose main characteristic is that it automatically boots on Kodi and you can't touch system files to avoid "breaking it".

Since this is my first article, I wanted to introduce what I've done or am doing. On future articles, I will develop more about those and other projects as well. In the next months, I want to build a "Alexa kind device", and also a IoT device using different sensors. I hope you have liked this experience and are willing to learn as I am doing. Please comment and share your questions, ideas, experiences, and I will be happy to read them!

About the Author

Manuel Daza is passionate about technology and of course human relationships, that is what brought him to a sales role into an IT company. After 10 years of career, he has developed a high knowledge in sales and negotiation, which combined with his background in International Marketing and Communications, helps him to better understand the full process of the sale.
You will find his thoughts on his YouTube channel: https://www.youtube.com/user/mdabarb

UNIX BLOG PRESENTATION

hubertf's NetBSD Blog

Dr. Hubert Feyrer

His studies of technical computer science at the University of Applied Sciences (FH) Regensburg were followed by employment as a Unix/Solaris/NetBSD system administrator and as a computer science teacher, both at the University of Applied Sciences Regensburg as well as Stevens Institute of Technology in Hoboken, NJ, USA. After receiving a Ph.D. in Information Science from the University of Regensburg start as developer of hard- and software as well as network and security solutions, with next promotion to IT manager (CTO). As such, performing security consulting according to ISO 27001 and later changing into the automotive sector as a Chief Information Security Officer (CISO). Recent occupations include working as CISO for one of Germany`s largest process- and people service provider, currently responsible as CISO for a major German car manufacturer.

For more information, visit http://www.feyrer.de/

Some time ago, I tried to solve some real-world geocaching/math problem. In my brute-force approach, I allocated a number of jobs on multiple-CPU machines rented from Amazon AWS and were running NetBSD/Xen. After doing so, I discovered that the load distribution was not utilizing all CPUs. NetBSD's processor sets were a first attempt to a workaround, but I wanted a proper fix, as this was apparently not specific to the Xen port but affected the NetBSD scheduler on all platforms. NetBSD developer Michael van Elst hinted me at the real problem, and he also provided the first patch to the problem. I did setup a test environment and ran some tests, and documented my findings in what is one of my favorite blog posts:

Learning More About the NetBSD Scheduler (... Than I Wanted to Know)

I had another chat with Michael on the scheduler issue, and we agreed that someone should review his proposed patch. Some interesting things arose from the discussion:

I learned a bit more about the scheduler from Michael. With multiple CPUs, each CPU has a queue of processes that are either "on the CPU" (running) or waiting to be serviced (run) on that CPU. Those processes count as "migratable" in runqueue_t. Now and then, the system checks all its run queues to see if a CPU is idle, and can thus "steal" (migrate) processes from a busy CPU. This is done in sched_balance().
The "stealing" (migration) has a positive effect in that; the process doesn't have to wait to get serviced on the CPU it's currently on. On the other side, migrating the process affects both CPU's data and instruction caches. Therefore, switching CPUs shouldn't be taken too easy.

If migration happens, then this should be done from the CPU with the most processes that are waiting for CPU time. In this calculation, not only should the current number be counted in but also a bit of the CPU's history ishould be taken into account. So processes that started running on a CPU are not again taken away immediately. This is what is done with the help of the processes currently on migratable (r_mcount) and some "historic" average. This "historic" value is

taken from the previous round in r_avgcount. More or less weight can be given to this, and it seems that the current number of migratable processes had too little weight over all processes to be considered.

What happens in effect is that a process is not taken from its CPU, left there waiting, with another CPU spinning idle. Which is exactly what I saw in the first place.

Also, what I learned from Michael was that there are a number of sysctl variables that can be used to influence the scheduler. Those are available under the "kern.sched" sysctl-tree:

```
% sysctl -d kern.sched
kern.sched.cacheht_time: Cache hotness time (in ticks)
kern.sched.balance_period: Balance period (in ticks)
kern.sched.min_catch: Minimal count of threads for catching
kern.sched.timesoftints: Track CPU time for soft interrupts
kern.sched.kpreempt_pri: Minimum priority to trigger kernel preemption
kern.sched.upreempt_pri: Minimum priority to trigger user preemption
kern.sched.rtts: Round-robin time quantum (in milliseconds)
kern.sched.pri_min: Minimal POSIX real-time priority
kern.sched.pri_max: Maximal POSIX real-time priority
```

The above text shows that much more can be written about the scheduler and its whereabouts. However, this remains to be done by someone else (volunteers are welcome!).

Now, while digging into this, I also learned that I'm not the first to discover this issue; and there already exists another a PR on this. I have opened PR kern/51615, but there is also a kern/43561. Funny enough, the solution which has been proposed there is about the same, though with a slightly different implementation. Still, *2 and <<1 are the same as /2 and >>1, so there is no change. And renaming variables for fun doesn't count anyways. ;) Last but not least, it's worth noting that this whole issue is not Xen-specific.

Thus, with this in mind, I went to do a bit of testing. I had already tested running concurrent, long-running processes that did use up all the CPU they got, and the test was good.
To test a different load on the system, I started a "build.sh -j8" on a (VMware Fusion) VM with 4 CPUs on a MacBook Pro. That nearly brought the machine to a halt - Though, what I saw was lots of idle time on all CPUs. I aborted the exercise to get me back some CPU cycles. I blame the VM handling here, not the guest operating system.

I restarted the exercise with 2 CPUs in the same VM, and there I saw load distribution on both CPUs (not much wonder with -j8), but there were also quite some idle times in the 'make clean / install' phases, which I'm not sure if it is normal. During the actual build phases I wasn't able to see idle time, though the system spent quite some time in the kernel (system). Example top(1) output:

```
load averages:  9.01,  8.60,  7.15;               up 0+01:24:11      01:19:33
  67 processes: 7 runnable, 58 sleeping, 2 on CPU
  CPU0 states:  0.0% user, 55.4% nice, 44.6% system,  0.0% interrupt,  0.0% idle
  CPU1 states:  0.0% user, 69.3% nice, 30.7% system,  0.0% interrupt,  0.0% idle
  Memory: 311M Act, 99M Inact, 6736K Wired, 23M Exec, 322M File, 395M Free
  Swap: 1536M Total, 21M Used, 1516M Free


  PID USERNAME PRI NICE   SIZE   RES STATE      TIME   WCPU    CPU COMMAND
  27028 feyrer    20    5    62M   27M CPU/1      0:00  9.74%  0.93% cc1
    728 feyrer    85    0    78M 3808K select/1   1:03  0.73%  0.73% sshd
```

```
23274 feyrer    21    5    36M   14M RUN/0      0:00 10.00%  0.49% cc1
21634 feyrer    20    5    44M   20M RUN/0      0:00  7.00%  0.34% cc1
24697 feyrer    77    5  7988K 2480K select/1   0:00  0.31%  0.15% nbmake
24964 feyrer    74    5    11M 5496K select/1   0:00  0.44%  0.15% nbmake
18221 feyrer    21    5    49M   15M RUN/0      0:00  2.00%  0.10% cc1
14513 feyrer    20    5    43M   16M RUN/0      0:00  2.00%  0.10% cc1
  518 feyrer    43    0    15M 1764K CPU/0      0:02  0.00%  0.00% top
20842 feyrer    21    5  6992K  340K RUN/0      0:00  0.00%  0.00% x86_64--netb
16215 feyrer    21    5    28M  172K RUN/0      0:00  0.00%  0.00% cc1
 8922 feyrer    20    5    51M   14M RUN/0      0:00  0.00%  0.00% cc1
```

All in all, I'd say the patch is a good step forward from the current situation, which does not properly distribute pure CPU hogs, at all.

MEET Dr. Hubert Feyrer

Can you tell our readers about yourself and your blog?

Sure! I'm Hubert Feyrer and I encountered NetBSD while studying computer science. Back then, I came from the Amiga, and lots of people then ported Unix software to the Amiga. There was a "real" Unix available for the Amiga from Commodore, but it was an expensive commercial software. A group of enthusiasts tried to port Mach, but theirs was a little progress. Then one morning, Markus Wild turned up with a port of NetBSD to the Amiga, and that got the ball rolling - back in 1993. I stuck around at the University of Applied Sciences in Regensburg for quite some time, switching between studying, working at the CS department, working on my PhD and more work at the CS department. This also gave me lots of room to work on NetBSD. My blog at http://www.feyrer.de/NetBSD/blog.html is (as you may guess) about NetBSD. The first posts go back to 2004, and the link collection that it started from and that is still available go back to 1996.

How you first got involved in blogging?

When I was in academia and Linux started to grow big, attracting people long after NetBSD was popular in the (comparatively) tiny Amiga scene/ Then, dedicated Open Source / Linux events became en vogue. When I was invited to many of these events, I showed people what NetBSD was, and back then when the Internet came into fashion, I also put up some early NetBSD-related web pages - e.g. the postcard that I got in 1993 thanking me for NetBSD help. Things started out as a list of URLs of my own and other people, and this evolved into my NetBSD blog. See http://www.feyrer.de/NetBSD/ for that page which still serves as today's visual template for my blog.

At first, everything was home-written software. Later, I switched to Blosxom for its simplicity. I know there are a lot better and newer software, but Blosxom fits my purpose pretty well. One very nice feature of Blosxom is to add tags to postings, and I maintain a tag cloud of topics related to NetBSD - see the bottom of http://www.feyrer.de/NetBSD/blog.html. I had the idea to work this into a documentation system, but this never got beyond adding a few standard tags - see http://www.feyrer.de/NetBSD/bx/tags.html.

I use homegrown software for tracking progress on my blog and other selected pages, and I'm happy to see that it's still pretty highly frequented, even though my time for NetBSD has decreased somewhat after leaving academia. Today's access count is about 5.000 to 6.000 hits on my entire blog. My blog is also one of those aggregated at http://netbsd.fi,

and I'm always happy to add small blog posts anytime, knowing that it will merge into the bigger stream of NetBSD-related content there.

What's the best thing a blogger can give to his readers?

Insight and wisdom. Many of my blog posts are there to index external URLs with tags for easier retrieval via the tag cloud. But I really enjoy when I can gain new insights myself, pass on that wisdom and see people get back to me on that.

Everyone has a favorite/least favorite post. Name yours and why?

Least... besides many things, I'm the author of g4u, a hard disk cloning software (see http://www.feyrer.de/g4u). Some time ago, the core code was taken by someone and incorporated their Linux-based derivative under GPL, without giving any credits. I did an analysis of this and shared the findings in my blog post as a documentation of the case. This example of my BSD licensed code being relicensed without my consent was unpleasant, and it got me to join many discussions. If you want a URL to look into this mess, start here:
http://www.feyrer.de/NetBSD/blog.html/nb_20040917.html

Favorite... wow, so many! We're talking about 13 years. Maybe one from the recent past: while solving a geocaching/math problem, I stumbled across a problem in NetBSD's scheduler, and worked with people from the NetBSD community to understand the issue, get a proposed fix. Actually, get that one tested and into the tree, and then adding proper documentation. This was spread out over several blog posts (plus some NetBSD problem reports and mailing list postings). In the end, the posting with the (my!) core enlightenment is this one: ``Learning more about the NetBSD scheduler (... than I wanted to know)'', URL:

http://www.feyrer.de/NetBSD/blog.html/nb_20161113_0122.html

This story even made it to BSDtoday, a BSD-focused webcast. See
http://www.feyrer.de/NetBSD/blog.html/nb_20161124_2128.html :-)

What reason is there to choose Unix over another operating system from a programmer perspective?

That's a broad question. What context do we watch this in - Windows and its whole Sharepoint / Excel / Visual-Whatever ecosystem? I'm afraid few people trapped in that universe either have an idea about the beauty and simplicity of Unix, or (probably more likely) they don't have a choice on the operating system they prefer to use. Greetings from the corporate world! :-) If you start out without any prior constraints, there are again many modern systems that are completely agnostic to the underlying operating system, in the web page business. There is a huge ecosystem around PHP, Ruby and Javascript that work pretty well on any Unix system, but also (more or less) on Windows. Coming from the former, the preference of Unix for these platforms is obvious. As an exercise, try automating SSH from your favorite scripting language on a Unix platform, and then see if it still works on Windows. Good luck! Going even deeper, leaving out all those existing frameworks with their layers of complexity added on top of the operating system: an operating system is a software which gives an abstraction to hardware. It comes with an interface to talk to the Unix shell and the C API in our favorite case. Those are simple and elegant, and can be combined to do many jobs. Like all those frameworks that make so many wonderful things possible - all those Unix-based Web Pages, desktop environments like GNOME and KDE, databases, and lots of hardware where you don't even know, there is Unix inside.

Of course, there is quite some insight in the "Unix" camp today with major players like Linux and the ones in the BSD camp. And while they all have great features of their own, what made them great are the simple concepts that come with Unix. At this point, I'd like to recommend an excellent book for those interested in the original Unix concepts, ``The UNIX Programming Environment'' by Brian W. Kernighan and Rob Pike.

The NetBSD 7.1 was released a few days ago. Can you tell us what the best change is if we compare it to the previous releases?

It has all the greatness of a major NetBSD release (NetBSD 7) but without the bugs - as it's an update to the NetBSD 7.0 release. Seriously, 7.1 comes with many new and great features, while at the same time it uses the NetBSD 7.0 codebase from the stable release branch. This does not add all the latest features (and bugs) in development, but only those that have ripened to a level of maturity, where developers took the labor to port them from the development (-current) branch to the netbsd-7 release branch. As a result, you can not only see a solid number of great new features in NetBSD 7.1 but also a very long list of security improvements and bugs that are fixed, making 7.1 the ideal base for (continued) long term support.

Please, see the NetBSD 7.1 release notes for all the details: http://www.netbsd.org/releases/formal-7/NetBSD-7.1.html (geez, I didn't manage to blog about this. Shame on me!)

What do you think is the future of NetBSD?

The goal of the NetBSD project is to provide a free, secure and portable Unix-like Open-Source operating system. I think this sums it up pretty well - NetBSD comes from early BSD, has evolved a lot over time while being true to itself, and I think we will see what the greater "Unix" universe expands to, and remain compatible. At some points, new features are adapted either because they prove to be good and mature (think threads, SMP - nothing that was there either in 1969's Unix nor in 1993's BSD!) or are required from a User perspective (like all those Linux ABI calls to run proprietary Linux software). Another part of NetBSD's feature is our great system for cross-compiling the whole system from many source platforms, and build code for many target hardware platforms. With more and more embedded and "Internet of Things" systems, this will become more and more important - and we have all this from a single source tree today. Thirdly, with NetBSD's great base as an operating system, we can add a lot of freely available software on top of it - and for this, we have our great 3rd party software system, pkgsrc.

Please tell us more about the projects you are involved with?

Currently, there are no specific projects I am involved in. I currently have less time for NetBSD than I want. During that time, I follow the project and help out in various positions - e.g. assisting in projects like Google's Summer of Code, monitor internal and external news source and communication of those that I feel are relevant to the NetBSD community. For this, I plan to continue using my blog. But I also got access to post in the name of The NetBSD Project on Facebook, so we will see what happens. :-) As for projects in the past, pkgsrc is alive and kicking, working on the next quarterly stable release. g4u is sort of sleeping right now, lacking time - yet I still feel strong for it, and I should find time to revive it, getting it close to NetBSD-current and do regular releases again. Maybe I will make this my next project, again!

Thank you

Scheduler-Series From My NetBSD Blog

(URLs old to new)

[20161105] NetBSD 7.0/xen scheduling mystery, and how to fix it with processor sets — http://www.feyrer.de/NetBSD/blog.html/nb_20161105_1754.html

[20161109] Looking at the scheduler issue again (Updated) — http://www.feyrer.de/NetBSD/blog.html/nb_20161109_0059.html

[20161113] Learning more about the NetBSD scheduler (... than I wanted to know) — http://www.feyrer.de/NetBSD/blog.html/nb_20161113_0122.html

[20161222] Bringing the scheduler saga to the finishing line — http://www.feyrer.de/NetBSD/blog.html/nb_20161222_2113.html

[20170109] Documenting NetBSD's scheduler tweaks — http://www.feyrer.de/NetBSD/blog.html/nb_20170109_2108.html

Dr.WEB®
CureIt!™

EMERGENCY CURING

for Windows workstations and servers

including those running other anti-virus software

FUNCTIONS:

- Cures Windows workstations and servers.
- Verifies the quality of the anti-virus software currently in use.

FEATURES:

- Dr.Web CureIt! doesn't require installation and doesn't conflict with any known anti-virus; consequently there is no need to disable the anti-virus currently in use to check a system with Dr.Web CureIt!.
- Improved self-protection and an enhanced mode for more efficient countermeasures against Windows blockers.
- Dr.Web CureIt! is updated at least once an hour.
- The utility can be launched from removable media including USB storage devices.

LICENSING FEATURES:

The utility is available for free when used for non-business purposes.

Interview with Kalin Staykov

Kalin Staykov is developer & DevOps engineer at one of the leading semiconductor manufacturers in Europe. He's leading a team of engineers focused on Unix based technologies aimed at building high amount of code for multiple architectures. His day-to-day work involves tool integration and support. In his spare time he often enjoys his Ham Radio hobby and his love for Photography.

The Future of Unix Operating Systems

I believe that Unix is here to stay – it is the one operating system which was born during the early days of computing technology and it's growing ever since. Nowadays, all Enterprise systems heavily rely on the stability and reliability it can offer. It is flexible, easy to adapt and enhance. It can be molded to meet almost every need.

I see some new trends that put automation above the basic idea of Unix being straightforward and easy to understand. I see complex dependencies forming out of the desire to make up for common limitations. We are adapting to new ways of work, and I don't hide that I'm in favor of keeping it simple. I enjoy being able to easily see what is under the hood and read the script or trace an error back to its source code. The effort I have to put in troubleshooting is increasingly higher.

Virtualization technologies changed the world for the better and yes – all sorts of clouds are forming from it. Decentralized services are the new mainstream, and our thoughts are bent under the pressure of disruptive marketing which promotes the passion for being online and using “The Cloud”. I see several layers of Unix technologies supporting the foundation of this new mainstream. This change has a high impact on the way we administer our systems.

I used to be very protective of my servers. Back in the days, the OS was bound to one hardware piece that was the holy ground for all operations. I made backups of it, I kept checking hardware state, and wrote automated monitoring to easily spot even small fluctuations in the way my systems work. Now everything changed, and it's more likely that I'll put on a script to bring several new software containers out of my few hyper-visors rather than thinking about what went wrong with the system that crashed. Is this wrong? Maybe. It's a fast fix, so we go for it – that's how we work today under the pressure of delivering more and more.

Technology is subject to constant change, and I see some good things happening. Still, I can't forget how beautiful Unix systems were destroyed by the business. My hope is that those past mistakes are not repeated and that there will

always be systems like BSD which are not hampered by the business. They should be the keepers of the technology in its purest form.

Can you tell our readers about yourself and your role nowadays?

I'm Kalin, and I'm the IT guy. OK, I guess that was how people knew me about 15 years ago when IT was still fresh, and people like me were an oddity. I was 15 years old when Internet was starting to become popular, and we had our first touch to it via dial-up phone modems. It was weird and time-consuming. A year later, I was introduced to Linux on a system that had only 8 MB of RAM. I recall that it took about one whole day to compile a kernel.

The first time I touched a keyboard was on an 8-bit system, if we don't count the old school typewriter. I wrote my first program at the age of 14, and it did not print "hello, world". I was a system administrator for many years, and I concentrated on Sun Solaris systems mostly. Nowadays, I consider myself a DevOps engineer on the good days and a bad programmer on the occasional bad day.

How you first got involved with programming?

I was staring at a greenish monitor. There was a prompt on the screen, and I asked a friend of mine "what now?". He introduced me to graphics, which involved writing three pages of Basic code just to see few big squares on the screen.

Can you tell us more about what you think about DevOps?

I think that the concept of DevOps is misunderstood by many. Some managers see it as the guy who is both an administrator and a programmer. Others think they are those smart guys, who put their whole infrastructure in code and orchestrate clusters of intelligent units, which run in a Cloud. Frankly, I don't dive into too many thoughts around concepts. In the end, we are just the Unix guys - we feel our way through the system via its shell and we don't ask too many questions. I've also heard about Windows DevOps engineers, but I don't have any evidence of their actual existence.

What is your the most interesting programming issue and why?

I wrote a small program that shows the meaning of life. But it never compiled.

Please, tell us more about your current projects?

I'm very much interested in Ruby and the Rails framework. I enjoy writing for web and drilling the web via semi-automated processing/crawler functions. There is so much information out there nowadays, and I enjoy "fishing" through it via code.

Many of our readers started programming journey. Do you have any tips for them?

Write some junk of code right at the beginning and try to have fun. Then, take a step back and learn the actual concepts of programming. If you do that right, it doesn't matter what language you'll pick and what your objectives are, all will go smoothly. Pick an actual project as soon as you can - learning is easier when you have a target and a strict deadline.

What is your favorite OS and why?

MacOS for its beauty and ease of use, Linux for its flexibility and BSD for style!

Do you have any specific goals for the rest of this year?

I placed too much on my plate already, so surviving does it for me. Building something really cool on Ruby on Rails is the secondary objective.

Thank you

Column

With the wounds still open after another heinous terror attack in the heart of London, calls are already being made that security services must be able to decrypt messages from the Facebook-owned service, WhatsApp. In light of the recent revelations of CIA back-doors in smart televisions, is this bluff, rhetoric, or a call for further a political clampdown on free speech?

Khalid Masood, the terrorist who slaughtered four innocent people last week, moments before he met his demise, sent an encrypted message which has prompted the Home Secretary to demand access to end-to-end encrypted messages by the government where necessary. While admitting encryption was essential to business and banking etc, this “floated kite” is allied to further pushes for Google *et al* to become content editors and screen out offensive content such as “fake news” etc. So the relentless behind the scenes of battle of who controls access, opinion and communications transmitted digitally over a bunch of wires or through the air continues unabated, while the bodies of the deceased are not yet resting in their graves. Amber Rudd may find end-to-end encryption “completely unacceptable”, but personally, I find such fruitless political opportunism and sound-bites even more so while families are grieving, and the injured coming to terms with has happened. Unfortunately these comments come over 70 years too late – in the case of German encryption and thousands of years in the case of the cruder Roman variety. It would appear a lot of people are really more scared of technology than of maniacs.

That politicians and investigative journalists are amongst the most active users of encryption (a fact that our Home Secretary conveniently forgets to mention) is a moot point. I don’t think the lady would be too impressed if all her correspondence appeared Hillary Clinton style on Wikileaks. I won’t even bring into the discussion the whole issue of dissidents from many nations that have managed to get their messages out. Nor will I rise to the bait of the old line “If you have nothing to fear, you have nothing to hide”. Such rhetoric reveals a naive confidence in justice, equality, and fairness. Just ask any politician or journalist.

This is what I believe. The lessons from World War II taught the Western nations much about the importance of signals intelligence, but more importantly, how crucial it is to infiltrate the other side and get the keys to the kingdom, by hook or by crook. As good quality encryption is pretty much a dark art on the same scale as quantum physics, alchemy, or casting out demons, the established order realized this and co-opted those involved. This would be an astute political move, as not only do you have plausible deniability but you can argue your support free speech, privacy, etc. while knowing damn well when push comes to shove you have the upper hand. Bluff, double bluff and counter bluff. Of course, I could be wrong, but on the balance of probabilities, I think I have a good argument here. The deep state will as a priority protect their interests, their secrets, and naturally, their assets.

On the other hand, Amber Rudd may have a point which will redress the cock-up rather than the conspiracy side of the argument. The government is notoriously disconnected from the forces that dominate the technology marketplace, slow to adapt to rapid changes in society. It may well be that Google, Facebook etc. do have the edge and the government is locked out. In which case, there are some sensitive questions to be asked – none of which will ever probably be answered satisfactorily for both commercial and security reasons. Firstly, do the major players have backdoors or access to encrypted data that they have not disclosed to the security services? If they don’t, does the author of the original encryption algorithm? Such matters place all the parties at major risk of legal and physical peril. If you are the only keyholder to an immense kingdom, you hold a phenomenal amount of power. Of course, the more people you share this with, the greater the risk that your security will be compromised. Secondly, is there an encryption algorithm out there that has not been secretly broken by some government somewhere? Again, in the land of subterfuge, misinformation and propaganda where the intelligence and secret services reside, this question will very likely remain honestly unanswered. Even if the

political will to demonize a political opponent was overwhelming, the commercial consequences to suggesting your opponent could perform this would be so huge the risk would be too great. And if as a government you could perform this sleight of hand, it would be in your best interests to keep very, very quiet about it. It is no wonder security researchers and coders tend to be a bit paranoid. The truth will be very hard to find, even less so to admit. It is like trying to nail jelly to a wall with a drawing pin.

In an ideal world, there would be no need for encryption, guns, armies, police officers or some would even say, politicians. To whom and the content of the message sent by Khalid Masood may never be known. Irrespective of this, what we should be aiming for is a transparent government and private communications (not the other way round) with the caveat that a judge or jury can authorize access to decryption where there is sufficient probable cause for investigation. Anything else is an overreach of authority.

As Albert Einstein said, “I know not with what weapons World War III will be fought, but World War IV will be fought with sticks and stones”. Encryption – like the nuclear arms race – follows the same cultural pathology. In the 1960’s, people were worried about the consequences of nuclear war with Russia, and the current generation is gradually waking up to the threats posed by globalism and the Internet at large. Even if the Home Secretary does manage to strong-arm the industry into opening the Pandora’s box of decryption on demand, tragically it will not stop the barbarism that occurred last week from continuing. Even if you banned sticks and stones, that intent on violence and harm would just use their hands.

About the Author

Rob Somerville has been passionate about technology since his early teens. A keen advocate of open systems since the mid-eighties, he has worked in many corporate sectors including finance, automotive, air- lines, government and media in a variety of roles from technical support, system administrator, developer, systems integrator and IT manager. He has moved on from CP/M and nixie tubes but keeps a soldering iron handy just in case.

MAGAZINE BSD

Editor in Chief:
Ewa Dudzic
ewa@bsdmag.org
www.bsdmag.org

Contributing:
Renan Dias, Rob Somerville, Hubert Feyrer, Kalin Staykov, Manuel Daza, Abdorrahman Homaei, Amit Chugh, Mohamed Farag, Bob Cromwell, David Rodriguez, Carlos Antonio Neira Bustos, Antonio Francesco Gentile, Randy Remirez, Vishal Lambe, Mikhail Zakharov, Pedro Giffuni, David Carlier, Albert Hui, Marcus Shmitt, Aryeh Friedman

Top Betatesters & Proofreaders:
Daniel Cialdella Converti, Eric De La Cruz Lugo, Radjis Mahangoe, Daniel LaFlamme, Steven Wierckx, Denise Ebery, Eric Geissinger, Luca Ferrari, Imad Soltani, Olaoluwa Omokanwaye, Radjis Mahangoe, Katherine Dizon and Mark VonFange.

Special Thanks:
Denise Ebery
Annie Zhang
Katherine Dizon

Senior Consultant/Publisher:
Paweł Marciniak

Publisher:
Hakin9 Media SK,
02-676 Warsaw, Poland Postepu 17D Poland
worldwide publishing editors@bsdmag.org

Hakin9 Media SK is looking for partners from all over the world. If you are interested in cooperation with us, please contact us via e-mail: editors@bsdmag.org

ServerU

Rack-mount networking server

Designed for BSD and Linux Systems

Designed. Certified. Supported

Up to **5.5Gbit/s** routing power!

KEY FEATURES

- 6 NICs w/ Intel igb(4) driver w/ bypass
- Hand-picked server chipsets
- Netmap Ready (FreeBSD & pfSense)
- Up to 14 Gigabit expansion ports
- Up to 4x10GbE SFP+ expansion

PERFECT FOR

- BGP & OSPF routing
- Firewall & UTM Security Appliances
- Intrusion Detection & WAF
- CDN & Web Cache / Proxy
- E-mail Server & SMTP Filtering

contactus@serveru.us | www.serveru.us
8001 NW 64th St. Miami, LF 33166 | +1 (305) 421-9956

IS AFFORDABLE FLASH STORAGE OUT OF REACH?

NOT ANYMORE!

IXSYSTEMS DELIVERS A FLASH ARRAY FOR UNDER $10,000

Introducing FreeNAS® Certified Flash. A high performance all-flash array at the cost of spinning disk.

KEY ADVANTAGES

- 10TB of all-flash storage for less than $10,000
- Unifies SAN/NAS for block and file workloads
- Runs FreeNAS, the world's #1 software-defined storage solution
- OpenZFS ensures data integrity
- Scales to 100TB in 2U
- Perfectly suited for Virtualization, Databases, Analytics, HPC, and M&E
- Performance-oriented design provides maximum throughput/IOPs and lowest latency
- Maximizes ROI via high-density SSD technology and inline data reduction

The all-flash datacenter is now within reach. Deploy a FreeNAS Certified Flash array today from iXsystems and take advantage of all the benefits flash delivers.

For more information, visit iXsystems.com/FreeNAS-Certified-Servers today.

iXsystems™